電子計算機の原理と構造

―ハードウェアの知識―

電気学会

電子計算機の原理と構造

ハードウェアの知識

執筆委員

日本電信電話公社　工学博士　桑原守二

電気学会

序　　言

電子計算機を中心とする情報処理技術の進展は，今後，ますます目ざましくなるであろう。したがって，情報化社会を推進する技術者・管理者を１人でも多く養成し，かつ，情報処理に関する正しい知識を普及することは，現下の急務である。

さきに，国においては，情報処理技術者認定制度を発足させ，また，高等学校の正式科目として情報処理教育を採り入れることを決めた。

電気学会は，終戦後間もないころより，わが国の電気産業の復興の一日も速からんことを願って，電気工学に関する講座を開設し，以来，多くの技術者を世に送って，技術者の養成に役立っている。さらに，今日の新たな情勢ならびに社会的要求に応えて，情報処理に関する一連の書籍を発行し，正しい，系統的な知識を，きわめて判りやすい形で普及することに努めることとした。本書はその一部をなすものである。

電気学会は，いまや，電気・電子工学のみならず，情報工学・制御工学・原子力工学等の新しい分野にも対応する総合学会としての機能を発揮し，多くの技術分野の学者・技術者の集団となっている。本会の発行するものが，その英知の結晶であることは，もとよりである。

特に，本シリーズの編修・執筆に当たっては，わが国電子計算機の生みの親ともいうべき，この方面の最高権威である山下，後藤両博士のご指導の下に，第一線の少壮技術者，学者，教育者よりなる編修委員会が組織され，その内容を慎重審議した。したがって，大学から高校にいたる各階層の学校の教科書ないし参考書，また社会人の生涯教育の手段として，あるいは認定試験を受ける人々の実力養成用として，本シリーズは，幅広く役立つことと思う。

終わりに，本事業の推進に尽力された本会歴代の役員各位ならびに編修・執筆その他，本書の製作に当たられた各位のご努力に対し，この場を借りて，厚くお礼申し上げる。

昭和45年８月

編修委員長　工学博士　尾　本　義　一

電子計算機部門編修委員

(主査)	東洋大学	工学博士	山下英男
	北豊島工業高校		緒方興助
	日立製作所		太田文平
	小金井工業高校		海谷虔郎
	日本電信電話公社	工学博士	桑原守二
	明治大学	工学博士	後藤以紀
	東海大学		佐藤達男
	商業教育共同実習所		篠原靖市
	蔵前工業高校		関口三郎
	川口工業高校		原田進
	東京理科大学		三重野博司

電気学会通信教育会編修委員

委員長	工学博士	尾本義一		工学博士	法貴四郎
幹事		巽良知		工学博士	星合正治
委員	工学博士	一本松珠璣			三ツ井新次郎
		井上清二		工学博士	森寿五郎
		柴田二三男		工学博士	山口次郎
		瀬戸千秋		工学博士	山下英男
		豊島嘉造		工学博士	山田直平
	工学博士	埴野一郎			吉岡俊男
		原田久			吉田確太
	工学博士	福田節雄		工学博士	六角英通

凡　　例

1. 用語は，日本工業規格（ＪＩＳ）ならびに文部省制定のものに従うのを原則とした。ただし，本書編修の時点においては，ＪＩＳ情報処理用語は未制定であるので，その原案を参考とした。

2. 図記号は，日本工業規格制定のものに従うのを原則とした。

3. 重要な用語は，見出しのほか，本文中においても初出の際，太字をもって示した。また，その英語を脚注に示した。

4. 外来語の発音をカナ文字で表記した用語については，なるべく，単語の区切りに“・”記号をそう入して読みやすくした。たとえば，オンライン・リアルタイム・システム等。

5. ２進符号の１，０が，他の１，０と混同されやすい箇所においては，特に前者を“１”，“０”で示した。

6. 計算機の操作手順は①，②，……等で示した。

7. 量や変数を表わす記号は，斜体とした。たとえば，A, f等。

8. 単位記号は，立体とした。たとえばＶ等。

9. 図，表および式の番号は，章ごとに通し番号とした。

10. 図，表を文中で最初に引用する際は，その図番，表番を太字をもって示した。

11. 本文中の記述で，補足説明の必要と思われる事項は，脚注で説明を行なった。

12. 各章の終りには，その章で得られた知識を整理するために摘要を設け，さらに，適切な練習問題を掲げ，いわゆる自己検査できるようにしてある。

電子計算機の原理と構造

ハードウェアの知識

目　　次

第1章　電子式卓上計算機

1.1　電子式卓上計算機とその操作方法 …… 1

1.1.1　電子式卓上計算機の外観 …… 1

1.1.2　操作方法 …… 1

1.2　計算機の構成 …… 2

入力回路——入力変換回路——記憶回路——演算回路——
出力変換回路——出力回路

1.3　計算機での数値の表現 …… 4

1.3.1　2進法 …… 5

10進法から2進法への変換——2進法から10進法への変換

1.3.2　2進化10進法 …… 8

3増しコード——2 out of 5コード

1.4　2進演算 …… 10

1.4.1　加　算 …… 10

1.4.2　減　算 …… 13

1.4.3　乗　算 …… 16

1.4.4　除　算 …… 16

1.5　計算機回路 …… 17

1.5.1　記憶回路 …… 17

1.5.2　演算回路 …… 21

加算回路——減算回路——乗算回路——除算回路

1.5.3　入出力変換回路 …… 54

入力変換回路(符号器)——出力変換回路(解読器)

1.5.4　制御回路 …… 58

1.6　論理回路と論理代数 …… 60

1.6.1 論理回路 …… 60
論理積回路——論理和回路——否定回路
1.6.2 論理代数 …… 65
論理積と論理和——変換，結合，分配の定理——否定の定理
——ド・モルガンの定理
1.6.3 カルナウ図 …… 70
1.7 論理回路の構成と回路部品 …… 78
1.7.1 回路部品 …… 78
ダイオード——トランジスタ
1.7.2 論理回路の構成 …… 84
ダイオードによる論理回路——トランジスタによる論理回路——
フリップフロップ回路
1.7.3 集積回路 …… 95
1.8 摘 要 …… 97
1.9 練習問題 …… 100

第2章 小形電子計算機

2.1 概 説 …… 103
2.1.1 電子計算機の用語 …… 104
プログラム——情報とデータ——電子計算機におけるデータの
表わし方——ハードウェアとソフトウェア
2.1.2 電子計算機の構成と機能 …… 107
入力装置——記憶装置——演算装置——出力装置——制御装置
2.2 主記憶装置 …… 113
2.2.1 概 要 …… 113
2.2.2 磁心による記憶の原理 …… 114
磁心——記憶の原理——読み出しと書き込み
2.2.3 電流一致方式 …… 120
読み出し——書き込みあるいは再書き込みで“1”を書き込む
とき——書き込みあるいは再書き込みで“0”を書き込むとき
2.2.4 磁心記憶装置の構成 …… 124
磁心マトリクスとスタック——アドレス——スタックでの記憶
——検査の方法

2.2.5 磁心記憶装置の動作 …… 131
読み出し動作——書き込み動作 …… 132
2.3 補助記憶装置 …… 133
2.3.1 概 要 …… 133
2.3.2 磁性面上におけるデータの記録 …… 133
書き込みと読み出しの原理——記録の方式
2.3.3 磁気テープ装置 …… 140
データの書き込みと読み出し——磁気テープ上のデータの記録
——磁気テープ装置の構造
2.3.4 磁気ドラム記憶装置 …… 148
磁気ドラム——磁気ドラム上のデータの記録——磁気ドラム
記憶装置の構成
2.3.5 磁気ディスク記憶装置 …… 154
ディスク・パック——磁気ディスク・パック記憶装置の動作
2.4 入出力装置 …… 160
2.4.1 概 要 …… 160
2.4.2 カード読み取り機とカードせん孔機 …… 161
カード——カード読み取り機——カードせん孔機——けん盤
カードせん孔機とけん盤カード検孔機
2.4.3 テープ読み取り機とテープせん孔機 …… 167
紙テープ——テープ読み取り機——テープせん孔機
2.4.4 行印字機 …… 171
2.4.5 制御卓 …… 176
2.5 演算装置 …… 177
2.6 制御装置 …… 180
2.6.1 概 要 …… 180
2.6.2 プログラム …… 180
2.6.3 機械語の命令 …… 183
2.6.4 命令の記憶 …… 186
2.6.5 命令の取り出しと実行 …… 189
命令取り出し段階——命令実行段階
2.6.6 制御の方法 …… 193
仮定の設定——装置類の動作——$(0000)_8$ 番地の命令の読み出
しと実行——$(0001)_8$ 番地の命令の読み出しと実行——$(0002)_8$

番地の命令の読み出しと実行
2.6.7 周辺装置の制御 …… 201
2.7 摘 要 …… 205
2.8 練習問題 …… 210

第3章 高性能の電子計算機

3.1 電子計算機を利用するシステム …… 212
3.1.1 オンラインとリアルタイム …… 212
オンラインとオフライン——リアルタイムとディレィドタイム
——オンライン・リアルタイム・システムとその構成
3.1.2 一括処理(バッチ処理)システム …… 217
3.1.3 メッセージの交換を行なうシステム …… 218
3.1.4 データの内容をリアルタイム処理するシステム …… 221
3.1.5 タイムシェアリング(時分割)・システム …… 222
3.1.6 その他のシステム …… 226
3.2 高性能の電子計算機の機能 …… 228
3.2.1 高性能の電子計算機に対する要求 …… 228
3.2.2 割り込み …… 231
3.2.3 主記憶装置の割り当て …… 235
3.2.4 オペレーティング・システム …… 239
3.2.5 記憶保護 …… 241
3.2.6 通信制御 …… 242
3.2.7 電子計算機複合体 …… 244
3.3 データ伝送 …… 247
3.3.1 データ伝送の方式 …… 247
データ伝送用の符号——直流伝送と交流伝送——変調方式——
直列伝送と並列伝送
3.3.2 通信速度 …… 253
通信速度の尺度——周波数帯域と通信速度
3.3.3 通信方式 …… 255
2線式と4線式——単向通信方式——半二重通信方式——
全二重通信方式
3.3.4 伝送制御 …… 257

タイミング制御——誤り制御
3.4 端末制御 …… 258
3.4.1 端末装置の種類と構成 …… 258
3.4.2 キーボード・プリンタ …… 261
キーボード(けん盤部)——プリンタ(印字部)
3.4.3 光学マーク読み取り装置 …… 264
3.4.4 光学文字読み取り装置 …… 267
3.4.5 文字表示装置 …… 271
3.4.6 図形表示装置 …… 274
CRT表示装置(出力装置)——ライトペン(入力装置)
3.4.7 押しボタンダイヤル電話機 …… 277
3.4.8 紙テープ，紙カード，磁気テープを使用する端末装置 …… 279
3.5 摘 要 …… 280
3.6 練習問題 …… 282

第4章 電子計算機の歴史と今後の動向

4.1 電子計算機の歴史 …… 283
4.2 電子計算機の発達と今後の展望 …… 285
4.2.1 処理速度と記憶容量 …… 285
4.2.2 電子計算機と人間 …… 287
4.2.3 電子計算機の応用 …… 288
4.3 摘 要 …… 289
4.4 練習問題 …… 289

練習問題の答 …… 291
索 引 …… 295

第1章　電子式卓上計算機

電子計算機というと，一般に磁気テープ装置や印刷装置などをともなった大がかりな電子計算機が連想され，普通の会社の事務や，商店に使用されている電子式卓上計算機とはまったく別のものであると考えている人が多い。

しかし，計算機としての基本的な動作——演算を行なう——においては，この両者はまつたく同じものである。

この章では，最も簡単な電子計算機である**電子式卓上計算機**について，その原理と構成を述べ，計算機というものの基本的な考え方を説明する。

1.1　電子式卓上計算機とその操作方法

1.1.1　電子式卓上計算機の外観

電子式卓上計算機(1)の外観は**第1.1図**に示すようなものである。なお，計算機の外観，仕様はメーカごとにかなり異なるが，ここでは，基本概念を説明するのが目的であるので，基本的なボタン以外はすべて省略して記述してある。

1.1.2　操　作　方　法

この計算機の操作方法を具体的な例により説明する。

〔例 1.1〕　25＋34＝59

の計算を行なう場合

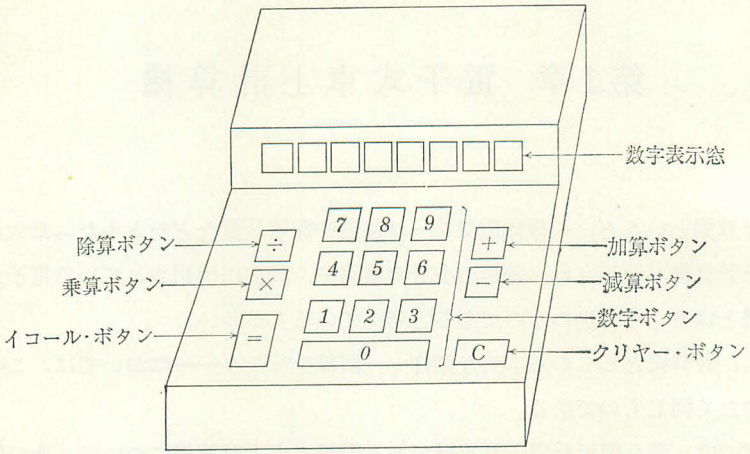
①　数字ボタンを[2]，[5]の順に押す。

数字表示窓に 25 が現われる。

②　演算ボタン[＋]を押す。

---

(1)　以下，特に電子式卓上計算機であることを強調したいとき以外は，計算機と略すことにする。



第1.1図　電子式卓上計算機の外観

数字表示窓は25のまま

③　数字ボタンを[3]，[4]の順に押す。

数字表示窓は34に変わる。

④　イコール・ボタン[=]を押す。

結果の59が数字表示窓に現われる。

他の演算についても同様に，

①　数字ボタン

②　演算ボタン

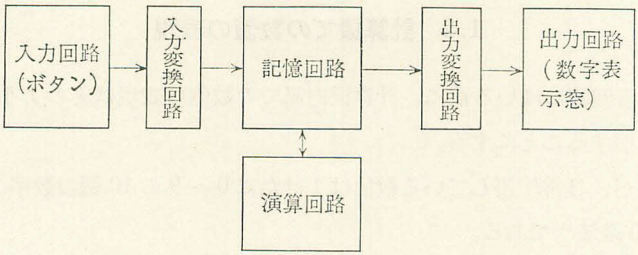
③　数字ボタン

④　イコール・ボタン

の順にボタンを押すことにより，計算が行なわれ，その結果が数字表示窓に現わされる。

1.2　計算機の構成

電子式卓上計算機といっても，その内部はかなり複雑なものであるが，基本となる回路構成をブロック・ダイヤグラムで示すと第1.2図のとおりである。



第1.2図 電子式卓上計算機の構成

各回路の詳細は後節で順次説明するが，ここでは，各回路の大まかな働きを説明する。

1. **入力回路**(1) 計算機で計算を行なうには，まず，計算しようとする数値と演算種別をなんらかの方法で計算機に指示してやらねばならない。この計算機に数値や演算種別を指示する回路を**入力回路**という。

電子式卓上計算機への数値の入力は，数字ボタンを押すことによって行なわれる。

2. **入力変換回路** 数字ボタンが押されたことを知り，そのボタンに対応する数値を計算機内部での処理に便利な符号（**コード**）(2) に変換する回路である。こうして変換された符号は**内部コード**といわれる。

3. **記憶回路**(3) 入力回路から与えられた数値や演算の途中結果等を一時的にたくわえ，記憶しておく回路である。記憶はすべて内部コードの形式で行なわれる。

4. **演算回路**(4) これは計算機の心臓部ともいうべき回路で，加，減，乗，除等の各種演算を行なう回路である。

5. **出力変換回路** 計算機内部での演算は，すべて内部コードにより行なわれるので，演算結果を数字表示窓に表示する必要がある。そのため内部コードを，該当する数字表示管を点火するための信号に変換する回路である。

6. **出力回路**(5) 入力回路とは逆に，演算結果の数値を人間に知らせるための回路である。電子式卓上計算機の場合，数字表示窓が**出力回路**に該当する。

---

(1) input circuit (2) code (3) memory circuit (4) arithmetic circiut
(5) output circuit

1.3　計算機での数値の表現

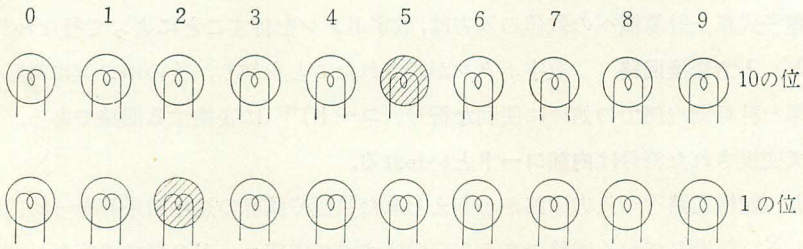
計算機の各回路にはいる前に，計算機内部での数値の表現はどのようになされているかを説明することにする。

われわれが，日常使用している数値は1けたに0～9の10個の数字を用いた，いわゆる**10進法**(1)である。

では，計算機において，この10進法をそのまま採用できるであろうか，例として52という数値を計算機内部で表現することを考えてみる。

簡単に考えて，この52という数値を表わす方法には2とおりあるであろう。

まず第一に考えられることは，各けたの10個の数値に1対1で対応する素子をおくことである。この場合，素子として小さなランプを考えると，1けたに10個の数値，2けたの数値を示すためには，2×10＝20 個のランプが必要である。



図は52を表わしている。素子の数20個必要
第1.3図　ランプによる数値の表現

この方法でさきの52を表示するためには，**第1.3図**で斜線を付したランプを点火してやればよい。この方法は，考え方はごく単純であるが，素子（この場合，ランプ）の数が非常に多くいることになる。一般に，N けたの数値を表現するには $N\times10$ 個の素子が必要となってくる。

第二の方法は，各けたに1個ずつの素子を使用し，その素子の状態を10段階に変えて，0～9の数値を表現する方法である。例を前と同じくランプにとってみると，このランプの明るさの状態を，たとえば，0→滅火，1→ほんの少し点火，2

(1)　decimal notation

→もう少し明るく点火，……，9→最高に明るく点火，といったように10段階に変化させる。しかし，この方法は，理論的には可能であるが，現実には安定度の点で実用的ではない。計算機では，まちがいを生ずることは何としても防ぐ必要があり，何にもまして素子の安定度，すなわち，ある状態と他の状態との区別のつきやすさが要求される。例で取り上げたランプを考えてみると，最も区別のつけやすい状態はランプがついているか，消えているかの2つの状態である。また，現在の計算機では，各回路の回路素子としてトランジスタが多く使われているが，このトランジスタにおいても，最も区別しやすいのは，やはり，トランジスタが動作しているか，動作していないかの2つの状態である。

こうした点から，10進法の数値を素子で表現するには，第一の方法がより望ましいということができる。

ここで，再びわれわれが日常無意識のうちに使っている10進法の性質をもう少し詳しく調べてみよう。

さきの10進数52は，これを書き変えてみると，

$$52=50+2=(5\times10)+(2\times1)=(5\times10^1)+(2\times10^0)$$

のようになる。また，3049という10進数は，

$$\begin{aligned}3049&=(3\times1000)+(0\times100)+(4\times10)+(9\times1)\\&=(3\times10^3)+(0\times10^2)+(4\times10^1)+(9\times10^0)\end{aligned}$$

となる。この例で示されている $10^0, 10^1, 10^2, 10^3, \cdots\cdots, 10^n$ の10を10進数の**基数**という。換言すれば，基数が10であるから10進法というのである。数の表現方法には種々の方法があり，基数が何であるかにより何進法かが決まる。

たとえば，基数が2であれば**2進法**(1)であり，この場合には，1けたに使う数字は，0，1の2個で，それ以上は上位へ**けた上げ**(2)となる。これは，10進法は0～9の10個の数字を使い，それ以上は上位へのけた上げが生ずることに対応している。また，基数が8であれば**8進法**(3)で，この場合は，各けたに0～7の8個の数字を使うことになる。

1.3.1 2 進 法

(1) binary notation (2) carry (3) octal notation

2進法の話をもう少し具体的にするために，はじめに例にあげた52という数を2進法で表わしてみると，

$$52=32+16+4$$
$$=(1\times2^5)+(1\times2^4)+(0\times2^3)+(1\times2^2)+(0\times2^1)+(0\times2^0)$$
$$=(110100)_2$$ (1)

さらに 3049 を2進法で表わすと，

$$3049=2048+512+256+128+64+32+8+1$$
$$=(1\times2^{11})+(0\times2^{10})+(1\times2^9)+(1\times2^8)+(1\times2^7)+(1\times2^6)$$
$$+(1\times2^5)+(0\times2^4)+(1\times2^3)+(0\times2^2)+(0\times2^1)+(1\times2^0)$$
$$=(101111101001)_2$$

第1.1表
10進法–2進法対応表

10進法	2	進	法		
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0

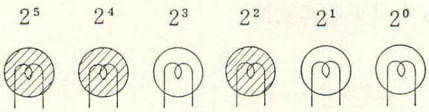
となる。このように，2進法は0と1の数値によって表現されるものである。第1.1表に0～20までの10進法と2進法の対応表をかかげてある。

ところで，さきにいろいろの素子で最も区別しやすい状態をとり得るのは，動作状態か不動作状態かの2とおりの状態であることを述べた。このことは，ある素子によって数値を表現しようとするには，数値の基数を2，すなわち，2進法を採用すればよいことになる。2進法では各けたの数値は0，1の2個のみであり，この0，1をたとえばランプの点滅あるいはトランジスタの動作(オン)，不動作(オフ)に対応させてやればよいことになる。

例として，さきほど第1.3図に示した52を2進数のランプで表示すると，第1.4図のようになる。この例でわかるように，同じ2値素子(2)（1個の素子が2つの状

(1)　10進数以外の数値を10進数と区別するため，$(\ )_n$ というような表現形式を使うことがある。この場合，2進数なので $n=2$ で $(\ )_2$ となっている。しかし，とくにまぎらわしくないときは $(\ )_n$ を省略してもよい

(2)　binary cell



第1.4図 ランプによる数値の表現

態しかとり得ないもの）を使っても，52という数値を表わすのに，10進法では20個必要であったが，2進法では6個の素子でよいことがわかる。

一般に，10進法の数値を2進法で表現するのに約3倍のけた数が必要である。また，10進法は1けたに10個の素子が必要とするため，結局，素子数は2進法の方が10進法のだいたい1/3でよいことになる。

このように2進法は，安定性の面からも素子数からもすぐれた体系であるので，現在のほとんどの計算がこの2進法で数値を表現している。

この0と1の数で表わされる2進数の各1けたを**ビット**(1)と呼んでいる。

この10進法，2進法の変換はつぎのように簡単に行なうことができる。

1. 10進法から2進法への変換 これは10進数を2進数の基数である2で割り，商と余りを求める。つぎに，その商を再度2で割り，また，商と余りを求める。以下つぎつぎに2で割り，商と余りを求めて行き，商が2より小さくなったところで計算をやめる。この最後の商を最上位けたとして，つぎに余りを下から順に並べれば求める2進数を得ることができる。

〔例 1.2〕 $(10)_{10}$ を2進数に変換

余り

$$\begin{array}{r|l} 2 & 10 \cdots\cdots 0 \\ 2 & 5 \cdots\cdots 1 \\ 2 & 2 \cdots\cdots 0 \\ & 1 \end{array} \quad \uparrow \text{並べる順序}$$

$\therefore\quad (10)_{10}=(1010)_2$

(1) bit: bit とは binary digit（2進数字）から作られた言葉である

〔例 1.3〕 $(52)_{10}$ を2進数に変換

		余り	
2	52	…… 0	↑
2	26	…… 0	|
2	13	…… 1	| 並べる順序
2	6	…… 0	|
2	3	…… 1	|
	1		↓

$\therefore \quad (52)_{10}=(110100)_2$

2. 2進法から10進法への変換　2進法の最下位のけたから順に 2^0, 2^1, 2^2, 2^3, ……, 2^N に対応させて（この 2^N を**重み**という），この重みにそのけたの数を掛けたものを全けた加算してやればよい。

〔例 1.4〕 $(1011)_2$ を10進数に変換

1	0	1	1
⋮	⋮	⋮	⋮
2^3	2^2	2^1	2^0

$$\therefore \quad (1011)_2=(2^3\times1)+(2^2\times0)+(2^1\times1)+(2^0\times1)$$
$$=(8\times1)+(4\times0)+(2\times1)+(1\times1)$$
$$=8+2+1$$
$$=(11)_{10}$$

〔例 1.5〕 $(1101011)_2$ を10進数に変換

$$(1101011)_2=(2^6\times1)+(2^5\times1)+(2^4\times0)+(2^3\times1)+(2^2\times0)+(2^1\times1)+(2^0\times1)$$
$$=(64\times1)+(32\times1)+(16\times0)+(8\times1)+(4\times0)+(2\times1)+(1\times1)$$
$$=64+32+8+2+1$$
$$=(107)_{10}$$

1.3.2　2進化10進法

いままで述べてきたことで，2進法の考え方と，なぜ計算機に2進法が採用され

ているかが理解できたことと思う。

ところで，われわれが日常で使っている数値は10進法であるため，計算機に数値を入れたり，計算結果を読み取ったりするには，10進法を使用しないと，きわめて不便である。このため計算機への入力，出力は10進法，計算機内部では2進法という2本立てをとらざるを得ない。このため，入力→内部，内部→出力のつど，それぞれ，10進法→2進法，2進法→10進法の変換がかならず必要となってくる。この10進↔2進変換は，かなり複雑な回路を必要とし，しかも，けた数が増えるにしたがってその複雑さも増大する。

ここで，この10進法↔2進法の変換を比較的容易に行なえ，しかも，2進法の特徴を生かすものとして，**2進化10進法**(1) がある。この2進化10進法は，10進法の各けたの数値を2進数で表わしたもので，たとえば52は，

$$\underbrace{0\,1\,0\,1}_{5}\quad\underbrace{0\,0\,1\,0}_{2}$$

となる。また，3459は，

$$\underbrace{0\,0\,1\,1}_{3}\quad\underbrace{0\,1\,0\,0}_{4}\quad\underbrace{0\,1\,0\,1}_{5}\quad\underbrace{1\,0\,0\,1}_{9}$$

となる。10進法の1けたは最大9までであるので，各けたは4ビットにより表示できることになる。

この2進化10進法は，変換は比較的容易であるが，あとで述べる計算時に少々面倒なことが出てくる。

本格的な計算機においては，演算速度が大きな問題点となるので，少々変換のわずらわしさがあっても，演算が早くできる純2進法を採用しているが，速度をそれほど要求されない電子式卓上計算機においては，変換の容易な2進化10進法を採用しているものが多い。

また，2進化10進法に類似するものに，**3増しコード**(2)，**2 out of 5コード**(3) があるが，ここでは，単に簡単な説明と第1.2表の対応表を示すにとどめる。

---

(1) binary coded decimal notation　(2) excess-three code
(3) two out of five code

第1.2表　10進法と各コードの対応表

10進法	2 進 法	2 進 化 10 進 法	3 増 し コ ー ド	2 out of 5 コード
0	0 0 0 0	0 0 0 0	0 0 1 1	1 1 0 0 0
1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 1 1
2	0 0 1 0	0 0 1 0	0 1 0 1	0 0 1 0 1
3	0 0 1 1	0 0 1 1	0 1 1 0	0 0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 1 1	0 1 0 0 1
5	0 1 0 1	0 1 0 1	1 0 0 0	0 1 0 1 0
6	0 1 1 0	0 1 1 0	1 0 0 1	0 1 1 0 0
7	0 1 1 1	0 1 1 1	1 0 1 0	1 0 0 0 1
8	1 0 0 0	1 0 0 0	1 0 1 1	1 0 0 1 0
9	1 0 0 1	1 0 0 1	1 1 0 0	1 0 1 0 0
10	1 0 1 0			
11	1 0 1 1			
12	1 1 0 0			
13	1 1 0 1			
14	1 1 1 0			
15	1 1 1 1			

1.　3増しコード　　2進化10進法に3を加えたもので，減算等で必要とする9に対する**補数**(1)を作るのに常に0と1を入れ換えるだけでよいのが利点である。

2.　2 out of 5コード　　このコードは0～9のどの数値も5ビットのうち1が2個あることが特徴で，これにより1が1個しかなかったり，3個あったりすると誤りであることがすぐわかるので，誤りの検出に便利である。

1.4　2 進 演 算

前項においては，計算機の数値表現が2進法（2進化10進法）でなされていることを述べたが，本項では，2進法による加，減，乗，除，すなわち，四則演算の方法について説明する。

1.4.1　加　算

10進法では，1けたが0～9の10個の数により表現されており，各けたの数値の和が9を越えるとけた上げを生じて，1つ上位のけたに“1”が加わる。

これに対して2進法では，1けたが“0”と“1”の2個の数により表現されており，したがって，あるけたの数が“1”を越えるとけた上げを生じ，1つ上位のけ

（1）　complement：たとえば，9に対する補数とは，加えると9になる数，たとえば3に対しては6，5には4，詳しくは1.4.2参照

たに“1”を加えることになる。用いられる数値が“0”，“1”の2種類であるため，2つの数値の和はつぎの4種類しかない。

```
  0      0      1      1
+) 0   +) 1   +) 0   +) 1
-----  -----  -----  -----
  0      1      1     1 0
                      ↑
                    けた上げ
```

ここで，2進数の加算例を示すと，

〔例 1.6〕

```
10 進 法      2 進 法
  3          0 0 1 1
+) 2       +) 0 0 1 0
-----      -----------
  5        0 1 0 1
             ↑
           けた上げ
```

〔例 1.7〕

```
  8          1 0 0 0
+) 5       +) 0 1 0 1
-----      -----------
 1 3       1 1 0 1
```

このように，2進法による加算はいたって簡単である。ここで，2進化10進法での加算を調べてみる。

まず，12+5=17 をみてみると，

```
10 進 法      2進化10進法
 1 2          0 0 0 1   0 0 1 0
+)  5      +) 0 0 0 0   0 1 0 1
-----      -------------------
 1 7          0 0 0 1   0 1 1 1
```

この場合はうまく結果が出ているが，つぎに12+9=21をみてみると，

```
 1 2          0 0 0 1   0 0 1 0
+)  9      +) 0 0 0 0   1 0 0 1
-----      -------------------
 2 1          0 0 0 1   1 0 1 1
```

上の例の答21は，0010　0001となるべきであるが，0001，1011となって正しい答が出ない。これは2進化10進法は各1けたを4ビットで表示しており，その4ビットのうち0000～1001を0～9に対応させ，1010～1111の6つのコードを使っていないことから生じるものである。したがって，2進化10進法の計算をする場合には，1010～1111のコードが結果として生じたときには，それを**補正**し，正しい結果が出るようにしてやらねばならない。この補正の方法にはいろいろあるが，ここでは，その1つ**±6補正**の方法について述べる。

±6補正　この補正の方法は，被加数の各けたにあらかじめ6を加えておいてから加算を行ない，もし，上位けたへのけた上げがあれば結果をそのまま使い，上位けたへのけた上げがない場合は結果から6を減ずるという方法である。この方法でさきの12+9=21を行なってみる。

①　被加数12の各けたへ6を加える。

	12	0001　0010
+)	66	0110　0110
		0111　1000

②　その数へ加数を加える。

		0111　1000
+)	9	0000　1001
		1000　0001
		けた上げなし　けた上げあり

③　けた上げのあったけたはそのままとし，けた上げのなかったけたから6を減ずる。

		1000　0001
−)	60	0110　0000
		0010　0001……結果

こうして，正しい結果0010　0001が得られる。

以上に示したように，2進化10進法による演算はかなり複雑な操作を必要としている。

他の減算や乗除算においても上記のような補正が必要となるのであるが，以後は

その補正方法の説明は省略する。

1.4.2 減 算

加算が理解できれば，減算も容易にわかるであろう。

0	1	1	1 0
−) 0	−) 1	−) 0	1
0	0	1	0 1

↑ 前のけたから1を借りる

例を示すと，

〔例 1.8〕

10 進 法	2 進 法
5	0 1 0 1
−) 3	−) 0 0 1 1
2	0 0 1 0

〔例 1.9〕

1 5	1 1 1 1
−) 9	−) 1 0 0 1
6	0 1 1 0

このように，10進法の考え方と同じ方法で減算が行なえる。しかし，計算機では減算も**補数加算**という加算の方法で行なっている。その理由は演算回路を加算回路と減算回路の両方を用意する必要がなく，加算回路と補数を作る回路があればよいからである。

では，この補数加算について説明する。説明を容易にするため，まず，例を10進法にとってみる。10進法における補数の考え方に2つある。1つは**10に対する補数**，もう1つは**9に対する補数**である。

最初の10に対する補数とは，N けたの数があるとき，

$$10\text{に対する補数}=(N+1\text{けたの最小数})-(N\text{けたの数}) \qquad (1.1)$$

たとえば，2の10に対する補数は8 (10−2=8)，348の10に対する補数は652 (1000−348=652) である。

つぎに，9に対する補数は同じくNけたの数があるとき，

$$9\text{に対する補数} = (N\text{けたの最大数}) - (N\text{けたの数}) \qquad (1.2)$$

2の9に対する補数は7 (9−2=7)，348の9に対する補数は651 (999−348=651) である。

この補数を使えば，減算を加算で行なうことができる。たとえば，減算

$$863-326=537$$

を考えてみる。この式は，つぎのように書き替えることができる。

$$\begin{aligned}863-326&=863+(1000-326-1000)\\&=863+\underbrace{(1000-326)}_{\substack{10\text{に対する補数}\\(674)}}-1000\end{aligned}$$

または

$$\begin{aligned}863-326&=863+(999-326-1000+1)\\&=863+\underbrace{(999-326)}_{\substack{9\text{に対する補数}\\(673)}}-1000+1\end{aligned}$$

この補数を使う計算をわかりやすく示すと，

10に対する補数による計算	9に対する補数による計算
8 6 3	8 6 3
+)　6 7 4……10に対する補数	+)　6 7 3……9に対する補数
1 5 3 7	1 5 3 6
↑無視	+)　└…→ 1
	5 3 7

となり，10に対する補数をとって計算する場合は結果の最高位のけたを無視し，9に対する補数をとった場合は結果の最高位けたに出てくる1を最下位けたに加えてやればよいことがわかるであろう。

では，この補数による加算の行ない方を2進法の演算にあてはめてみる。

2進法の場合の補数は，**1に対する補数**と**2に対する補数**があり，それぞれ，10進法の9に対する補数および10に対する補数に該当する。

いま，4けたの2進数1000を考えてみる。4けたの2進数の最大値は 1111，したがって4けたの2進数1000の1に対する補数は，0111 (1111−1000=0111) であり，2に対する補数は，1000 (10000−1000=1000) である。0～9までの2進数の1に対する補数，2に対する補数の対応表を第**1.3**表に示す。

第1.3表　2進数の補数

10進法	2 進 法	1に対する補数	2に対する補数
0	0 0 0 0	1 1 1 1	0 0 0 0
1	0 0 0 1	1 1 1 0	1 1 1 1
2	0 0 1 0	1 1 0 1	1 1 1 0
3	0 0 1 1	1 1 0 0	1 1 0 1
4	0 1 0 0	1 0 1 1	1 1 0 0
5	0 1 0 1	1 0 1 0	1 0 1 1
6	0 1 1 0	1 0 0 1	1 0 1 0
7	0 1 1 1	1 0 0 0	1 0 0 1
8	1 0 0 0	0 1 1 1	1 0 0 0
9	1 0 0 1	0 1 1 0	0 1 1 1

この補数を使って，7−3=4 の計算を行なうとつぎのとおりである。

2に対する補数による計算

```
    0 1 1 1……7
+)  1 1 0 1……3の2に対する補数
─────────────
   1 0 1 0 0    4
   ↑
   無
   視
```

1に対する補数による計算

```
    0 1 1 1……7
+)  1 1 0 0……3の1に対する補数
─────────────
   1 0 0 1 1
   ⋮
   └------→1
─────────────
    0 1 0 0……4
```

以上でわかるように，10進法の補数加算とまったく同様に，2に対する補数による計算の場合は，最上位けたの1を無視し，1に対する補数による計算では，最上位けたの1を最下位けたに加えてやればよい。

このようにして補数をとってやれば，減算を加算で行なうことができる。ところで，この補数加算を1に対する補数でやるか，2に対する補数でやるかが問題になるが，2に対する補数を用いれば，計算は一度でよいが補数の作成がやや面倒である。一方，1に対する補数は第1.3表からもわかるように，1と0を入れ換えることにより容易にできる。小形の計算機においては，この補数作成回路を簡単にするため，1の補数を使用しているものが多い。

1.4.3 乗　算

10進法の乗算，除算において，絶対に必要なものに掛算の九九がある。計算機において10進九九を行なうのはかなり大変であるが，しかし，2進法においては数字が0と1しかないので，きわめて簡単になる。すなわち，2進法の九九に相当するものは，つぎの4とおりしか存在しない。

$$0\times0=0$$
$$0\times1=0$$
$$1\times0=0$$
$$1\times1=1$$

あとは10進法の乗算の仕方とまったく同じようにけたをずらして掛け合わせ，加えればよい。

いま例として，$2\times5=10$ の計算を行なってみる。

〔例 1.10〕

10進法		2進法	
2		0 1 0	
×) 5		×) 1 0 1	
10		0 1 0	
		0 0 0	部分積
		0 1 0	
		0 1 0 1 0	

この例のように，きわめて簡単である。また，この演算例をよく見ると，各けたの乗算で生ずる**部分積**は，乗数が1の場合は被乗数（この場合 010）がそのまま現われ，乗数が0の場合は0となっていることがわかるであろう。このことは計算機で乗算を行なうときまことに便利である。

1.4.4 除　算

筆算での2進法の除算は，10進法の除算とまったく同じ方法を取り，減算とけたずらしの繰り返しである。

$9\div3=3$ の例を行なってみると，

```
          0 0 1 1……3
3……1 1 ) 1 0 0 1……9
           1 1
         ---------
             1 1
             1 1
         ---------
               0
```

計算機でもこの方法を取っているのであるが，ただ，商の立て方に問題がある。人の場合は，目でみることにより，そのけたに1が立つか否かすぐに判断できるが，計算機ではこの判断が簡単にできない。この判断の方法もいろいろと考えられるが，その1つの方法としては，被除数から除数を一度引いてみて，もし引けたら（差が正なら）商に1を立て，つぎのけたに進み，また，もし引けなかったら（差が負なら）商に0を立て，つぎのけたに進むというやり方である。上の例でいけば，まず，被除数の最上位けたの1から除数を引く計算1−11を行ない，その結果差が負になるので，商に0を立て，つぎのけたに進む。今度は10−11を行ない，これも差が負になるから商に0を立て，つぎのけたに進む。さらに，100−11を行ない，今度は引けるので商に1を立てて，つぎのけたに進んでいくといった手順となる。この方法の詳しいことは，あとの除算回路の項で改めて説明する。

1.5 計算機回路

いままでの説明で計算機内部での数値は2進法で表現されており，また，その2進法の演算は，どのような考え方で行なえばよいか理解できたことと思う。

これから，その計算機の内部で，数値がどのように記憶されているか，また，さきに述べた2進法の演算をやるための回路はどのような構成になっているかについて説明する。

1.5.1 記憶回路

いま，計算機で計算しようとする場合，まず，演算するための数値を計算機内に入れなければならない。

卓上計算機の場合，その数値はボタンを押すことにより計算機の中にはいるのであるが，たとえば，25＋34＝59 という計算を行ないたいときは，

① [2], [5] とボタンを押す。

② [+] ボタンを押す。

③ [3], [4] とボタンを押す。

④ [=] ボタンを押す。

の順でボタンを押す。この場合，計算機は最後の[=]ボタンが押され，演算を行なうまでの間，①で押された25と，③で押された34の2つの数値をなんらかの形で記憶しておかねばならない。

また，乗算等のように，部分積をとって計算するような場合は，その演算の途中結果をたくわえておく必要がある。このように数値を一時的に覚えておく記憶回路を**レジスタ**(1)という。

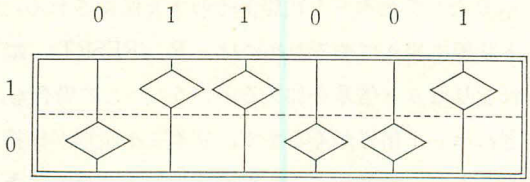
上の計算の例でいえば，被加数25を入れるためのレジスタ，加数34を入れるためのレジスタ，和の59を入れるためのレジスタと3つのレジスタが必要である。この3つのレジスタの状態は，**第1.5図**のように表わされる。

被加数レジスタ	0 1 1 0 0 1	(25が記憶されている)
加数レジスタ	1 0 0 0 1 0	(34が記憶されている)
和レジスタ	1 1 1 0 1 1	(59が記憶されている)

第1.5図　レジスタによる数値の記憶

さて，このレジスタがどのようなものであるかは，モデルとして1つ玉のそろばんを想像してもらえばよい。このそろばんは，玉が各けたに1つしかないので，各けたが取り得る状態は，玉が上にあるか下にあるかの2つのみである。いま，このそろばんの玉が上にあるときを“1”，下にあるときを“0”と考えれば，各けたの玉の配置を変えることにより，任意の2進数を表わすことができる。さきの例で挙げた被加数レジスタの内容 $25=(11001)_2$ を記憶させるには，各けたの玉の配置を**第1.6図**に示すようにすればよい。

(1) register



第1.6図 $(011001)_2$ を表わす1つ玉のそろばん

実際のレジスタは，このそろばんの各けたの玉を，ある回路素子におき換えたものである。この回路素子は，そろばん玉と同じ働き，すなわち，0と1の2つの状態を保持できるものであればよい。こうして，レジスタは電子的な1つ玉のそろばんと考えることができる。

このそろばん玉の働きをする回路素子はいろいろあるが，レジスタの構成素子としては，**フリップフロップ回路**(1)が広く使用されている。このフリップフロップ回

第1.7図 フリップフロップ回路の図記号

路を表わすシンボル（図記号）として第1.7図のものを使用する。この図で，S，Rは入力端子を，また，1，0は出力端子を示している。

こうしたフッリプフロップ回路は，そろばん玉の上下に対応するものとして，**セット**(2)状態，**リセット**(3)状態の2つの状態をとることができ，セット状態では出力端子1に電圧が現われ，また，リセット状態では出力端子0に電圧が現われる。

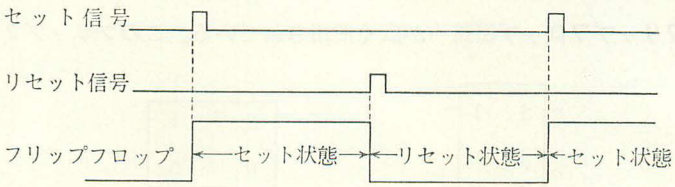
こうして，セット状態を“1”に，リセット状態を“0”に対応させて考えることができる。セット(1)，リセット(0)のどちらの状態をとらせるかは，入力端子S，Rのどちらに電圧が加えられるかにより決定される。すなわち，入力端子S (SET)に電圧を加える（これを**セット信号**を加えるという）と，このフリップフロップ回路はセット状態になり，一度セット状態になると，Sに加えられた電圧，すなわ

(1) flip-flop circuit (2) set (3) reset

ち，セット信号を取り去ってもセット状態がそのまま保持される。このセット状態を，反対のリセット状態に変えてやるためには，R（RESET）端子に電圧を加えてやればよい（これを**リセット信号**を加えるという）。この場合も，一度リセット状態になると，つぎにセット信号がくるまで，リセット状態が保持される。

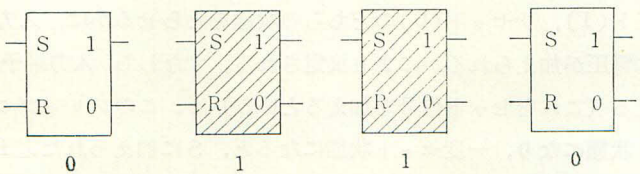
このことは，結果として一番最後にどちらの入力端子に信号がきたかを記憶していることになる。

これをそろばん玉に対応させて考えてみると，そろばん玉を上に押し上げる指の働きがセット信号であり，下におろす働きがリセット信号である。そしてそろばん玉を上または下に動かしたあとは指を放してもそろばん玉は動かされた状態のままになっている。



第1.8図　入力信号とフリップフロップの状態

この様子を時間的に示したものが**第1.8図**である。この図でわかるように，入力信号が切れてもその入力信号により与えられた状態を保っていることは，入力信号をそのまま記憶していることにほかならない。したがって，このようなフリップフロップをいくつか並べると，その個数に対応するけた数の2進数を記憶するレジスタを作くることができる。たとえば，4けたの2進数，すなわち，4ビットを記憶できるレジスタを作るには，このフリップフロップを4個並べればよい。そして，



第1.9図　4ビットのレジスタ（6を記憶）

このビットのレジスタに6を2進数で記憶させるには，**第1.9図**のように斜線を付したフリップフロップのS端子に信号を加えてやればよいことになる（ただし，記憶させる前に全部のフリップフロップをリセット状態にしておく）。

レジスタの大きさ，すなわち，何個のフリップフロップでレジスタを構成するかは，その計算機で扱う数値の大きさにより決定されることになる。

注　フリップフロップという言葉は，シーソーのようなもの，すなわち，ちょっとの刺激でその状態が反転するようなものを示している。なお，電子的なフリップフロップはセットまたはリセット信号によりその状態を反転させている。

1.5.2 演算回路

1. 加算回路　2進加算を筆算で行なう方法については，**1.4** 2進演算の節で述べたが，ここでは，まず，2進1けたの加算を行なう回路を考えてみる。

前述のように2進法における加算の組み合わせは，つぎの4とおりである。

$$0+0=0$$
$$0+1=1$$
$$1+0=1$$
$$1+1=0 \quad (\text{けた上げ}1)$$

これを，いま加算する2つの数を A，B，そのけたの結果（和）を S，けた上げを C として組み合わせ表を作成すると，**第1.4表**のようになる。この第1.4表のように，与えられた条件のあらゆる組み合わせに対して，その結果を示したものを**真理値表**(1)という。

ここで，2進加算のための加算回路を細かい回路動作は別にして，まずどのような働きをしなくてはならないかを考えてみる。加算回路と入力，出力の関係は**第1.10図**に示すとおりである。

第1.10 図　加算回路

(1) truth table

第1.4表　2進加算の組み合わせ

入力		出力	
被加数 A	加数 B	和 S	けた上げ C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

この加算回路の働きは，第1.4表の条件をすべて満足させるものでなければならない。すなわち，

Ⅰ．S が1となるための条件

①　$A=0$ であり，かつ，$B=1$ であるとき

②　$A=1$ であり，かつ，$B=0$ であるとき

Ⅱ．C が1となるための条件

$A=1$ であり，かつ，$B=1$ であるとき

加算回路は，このⅠ，Ⅱの条件を満足させる機能を備えたものであればよい。

a.　and 条件（アンド）　上記の条件のうち，最初に示した「$A=0$ であり，かつ，$B=1$ である」というような条件のつけ方を **and 条件**という。この and 条件は，「$A=0$ である」だけではだめであり，また，「$B=1$ である」だけでもだめで，あくまで，「$A=0$ であり，かつ，$B=1$ である」ことが必要である。

同じく，Ⅰの②の「$A=1$ であり，かつ，$B=0$ である」という条件，およびⅡの「$A=1$ であり，かつ，$B=1$ である」という条件も and 条件である。

この and 条件の例を日常の生活にとってみると，たとえば，ある町内の運動会を，天気の良い日曜日に行なうことに決めたとすると，運動会が開かれるのは，天気が良く，しかも日曜日だということで，天気が良いだけではだめであるし，日曜日だということだけでもだめで，あくまで日曜日であり，しかも，天気が良いということが必要となる。この場合，「天気が良い」ということと，「日曜日である」ことが運動会を開くための and 条件となっている。

この and 条件を簡潔に表示するために，つぎのような代数式に似た式を用いて

いる。すなわち，$A=1$，$B=1$，$C=1$ であることを，それぞれ，A，B，C と表わしたとき，IIの「$A=1$ であり，かつ，$B=1$ である」ときに C が1となるという条件を

$$C=A\cdot B \tag{1.3}$$

と書き表わす。ここで，C は A と B の**論理積**(1) (and) であるという。

この式の A と B の間に書かれている“・”は，「かつ」の意味を持っている。

つぎに，Iの②の S に1が出るための条件，$A=1$ であり，かつ，$B=0$ であるときの式は，

$$S=A\cdot B' \tag{1.4}$$

と書き表わす。このうち，S，A，・は前式と同じ意味を持つが，B' は，ここでは $B=0$ であることを表わすものとし，「Bの否定」と読む(2)。同様に，Iの①の条件を示す式は，

$$S=A'\cdot B \tag{1.5}$$

となる。

また，運動会を開くための and 条件は，

運動会の開催＝天気が良い・日曜日

と書けるであろう。

b. or 条件（オア）　ところで S に1が出るための式は，

$$S=A\cdot B' \text{ と } S=A'\cdot B \tag{1.6}$$

の2式により表わされている。このことは S が1になるためには，$A\cdot B'$ であるか，または，$A'\cdot B$ であるというどちらか1つの条件を満足すればよいことを示している。

このように2つ以上の条件があるとき，そのうちいずれか1つの条件を満足すればよいような条件を **or 条件**という。

これも，さきほどの運動会の例を引き，運動会の参加者は町内の者か，またはこの町内にある保育園に通っている者という条件をつければ，この町内の者と町内の保育園に通っている者とは or 条件になっている。すなわち，町内の者であるか，

(1) logical product, “and”　(2) B' を $\bar{B}$, $\sim B$, $\neg B$ などと書くこともある

あるいは，町内の保育園に通っている者であれば，どちらもこの運動会に参加できるのである。

この or 条件のいずれか1つあればよいということは，両方あってもよいことを含んでいる。この例では，町内に住み，町内の保育園に通っているものも当然参加資格があるのである。

この or 条件を表わすのには，“∨”(1) の記号を用い，S に1がでるための条件を示す式は，

$$S=A'\cdot B\vee A\cdot B' \tag{1.7}$$

と書かれる。この場合，S は $A'\cdot B$ と $A\cdot B'$ の**論理和**(2)（or）であるという。

運動会の例では，

運動会の参加者＝町内の者＋町内の保育園に通う者

と書けるであろう。

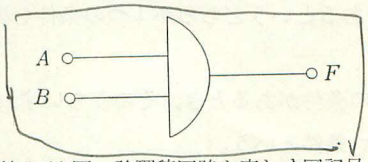
いままでに述べてきた and，or の式で，第1.4表の加算回路の真理値表を整理してみると，つぎのようになる。

$S=A'\cdot B\vee A\cdot B'$ ……和に1が出る条件　(1.8)

$C=A\cdot B$ ……けた上げが生じる条件　(1.9)

そこで，2進1けたの加算回路を作成するには，上の式に示される and 条件，or 条件を満足する回路を作り，これを組み合わせればよいことになる。

このうち，and 条件を満足する回路を**論理積回路**（**and 回路**）(3)，または**and ゲート**(4)といい，その図記号（シンボル）は**第1.11図**のように描かれる。



第1.11図　論理積回路を表わす図記号

この図記号は，A，B がこの回路への入力端子を，また，F がこの回路の出力端

（1）この“∨”という記号は，一般の加算の場合とはまったく意味が違うので注意しなければならない

（2）logical sum, “or”　（3）“and” circuit　（4）“and” gate

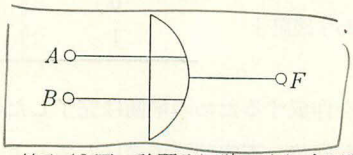
子を表わしており，入力端子 A，B の両方に 1 を示す記号がきたときのみ出力端子 F に 1 を示す信号を出す回路を表わしている，

この入力端子 A，B，出力端子 F の信号関係を表わす真理値表は，**第 1.5 表**のとおりである。

第 1.5 表　論理積回路の真理値表

入力		出力
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

つぎに，or 条件を満足する回路を**論理和回路**（**or 回路**）[1]，または **or ゲート**[2] といい，その図記号は**第 1.12 図**のように示すものである。



第 1.12 図　論理和回路を表わす図記号

この図記号は，入力端子 A，B のいずれかに（両方でもよい）1 を示す信号がきたとき，出力端子 F に 1 の信号を出す回路を表わしている。この論理和回路についても，論理積回路と同様な真理値表を示すと**第 1.6 表**のようになる。

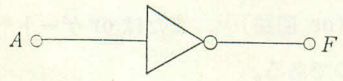
第 1.6 表　論理和回路の真理値表

入力		出力
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

(1)　"or" circuit　　(2)　"or" gate

これで，and ゲート，or ゲートの図記号を表わす意味は理解できたことと思うが，この and ゲート，or ゲートはすべて入力に1を示す信号がきたとき条件を満足するよう意味づけされているため，$A \cdot B'$ や $A' \vee B$ のように0信号を含む条件を満足できないことになる。

このため，信号を逆 (0→1, 1→0) にする機能を持つ**否定回路 (not 回路)**[1] がある。この否定回路は，**インバータ**[2]とも呼ばれている。否定回路の図記号は，第1.13図に示すものを使用し，入力端子 A に1を示す信号がくれば出力端子 F に0信号を，また A に0信号がくれば出力端子 F に1信号を出す回路を表わしている。この場合の入出力の関係を表わす真理値表は**第1.7表**のようになる。



第1.13図　not 回路を表わす図記号

第1.7表　否定回路の真理値表

入　力	出　力
A	F
0	1
1	0

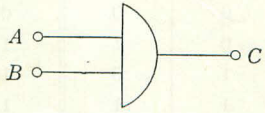
これで，2進加算回路を作成するための準備は完了したので，いよいよ2進加算回路を論理積回路，論理和回路，否定回路で構成してみよう。なお，論理積，論理和，否定回路をいかなる回路部品で作成するかは，後節**1.7**において説明する。

c.　**半加算器**[3]　　ここでもう一度加算のための式を示すと，

$$S = A' \cdot B \vee A \cdot B' \tag{1.10}$$

$$C = A \cdot B \tag{1.11}$$

まず，簡単な式 (1.11) $C = A \cdot B$ の方を and ゲートの図記号を用いて表わすと**第1.14図**のようになる。この図は，さき程説明したように，A, B の両方に1が



第1.14図　けた上げを示す and ゲートの図記号

(1)　"not" circuit　　(2)　inverter　　(3)　half-adder

きたときのみ C に1を出す回路を示しており，第1.4表のけた上げの生じる条件を満足することがわかるであろう。

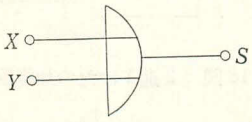
つぎに，和に1がでるための条件式 $S=A'\cdot B\vee A\cdot B'$ であるが，いま便宜的に

$$X=A'\cdot B,\qquad Y=A\cdot B' \tag{1.12}$$

とおくと，式 (1.10) は，

$$S=X\vee Y \tag{1.13}$$

と書くことができる。これを or ゲートを用いて表わすと，**第1.15図**のようになる。

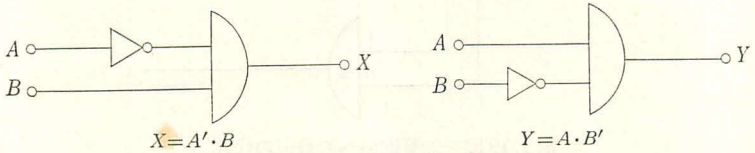


$S=X\vee Y$

第1.15図 和を示す or ゲート

そこで，この X, Y をもとにもどして，それぞれ and ゲート，否定回路を用いて表わすと，**第1.16図** (**a**)および同図(**b**)のようになる。

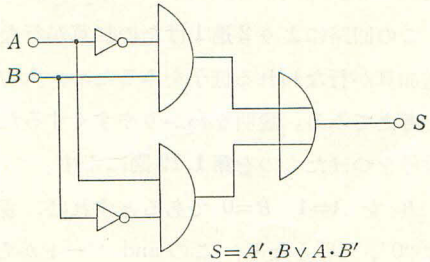
これを第1.15図の X, Y のところに代入すれば，**第1.17図**の回路ができ上がる。



$X=A'\cdot B$ $\qquad$ $Y=A\cdot B'$

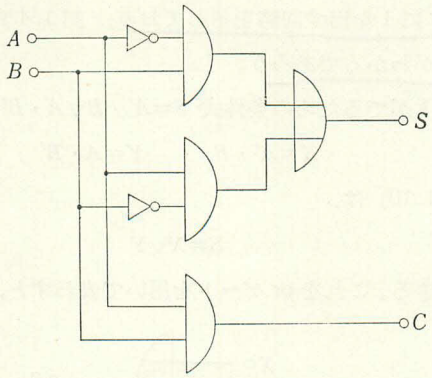
(a) X の条件 $\qquad$ (b) Y の条件

第1.16図

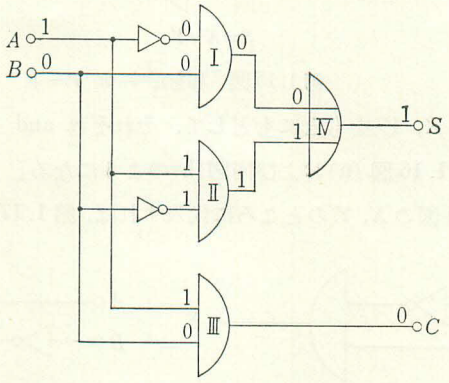


$S=A'\cdot B\vee A\cdot B'$

第1.17図 和を示す回路



第1.18図　2進1けたの加算回路（半加算器）



第1.19図　加算回路への数値の代入

これで式（1.10），式（1.11）を満足する回路ができたので，これをまとめて表わすと，**第1.18図**のようになる。この第1.18図が第1.4表の真理値表のすべてを満足する回路であり，この回路により**2進1けた**の加算が行なえる。

この回路により2進加算が行なわれる様子をみるため，入力 A，B に実際に数値を入れて回路の動作を考えてみる。説明をわかりやすくするため，第1.18図の加算回路の各ゲートに番号をつけたものを**第1.19図**に示す。

いま仮に，入力 A，B を $A=1$，$B=0$ であるとすれば，第1.19図のゲート**I**のandゲートの入力は“0”，“0”となり，このandゲートからの出力はない（“0”となる）。つぎのゲートIIの入力は“1”，“1”となり，and 条件を満足し，このゲ

ートの出力には“1”が出る。このようにゲートの入力条件を満足し，出力に“1”がでることをゲートが開くという。ゲートⅣの入力をみると，ゲートⅠからの出力は“0”，ゲートⅡからの出力は“1”であり，このゲートは or ゲートであるので，入力のうちいずれかに“1”がくるとゲートが開き出力の S が“1”になる。

また，ゲートⅢの and ゲートは，入力が“1”，“0”で条件を満足しないから，ゲートは開かれず，したがって，C に出力は出ない。結局，出力は $S=1$，$C=0$ となり，これは第1.4表の真理値表の3段目 $A=1$，$B=0 \rightarrow S=1$，$C=0$ の条件を満足している。他の組み合わせについても同様に条件を満足していることは，各自でこの回路を追って確かめていただきたい。

これまで述べてきた回路は，2進1けたの加算回路のうち最下位のけた，すなわち，下位からのけた上げがない場合の加算回路である。この下位からのけた上げを考慮に入れてない加算回路は，**半加算器**と呼ばれる。

d. 全加算器(1)　2進1けたの加算回路で，下位からのけた上げを考慮に入れたものを，半加算器に対し，**全加算器**という。

2つの数を加算する場合，最下位のけた以外は下位からのけた上げをつねに考えに入れて計算を行なわねばならないので，加算回路としては全加算器の方が一般的である。

この全加算器の入力としては，半加算器の入力 A，B の他に，下位からのけた上

第1.8表　全加算器の真理値表

No	入力			出力	
	A	B	C_1	S	C_2
1	0	0	0	0	0
2	0	1	0	1	0
3	1	0	0	1	0
4	1	1	0	0	1
5	0	0	1	1	0
6	0	1	1	0	1
7	1	0	1	0	1
8	1	1	1	1	1

(1) full-adder

げを入力として加えねばならない。いま，この下位からのけた上げを C_1 とすれば，この全加算への入力は，A, B, C_1 の3つとなる。

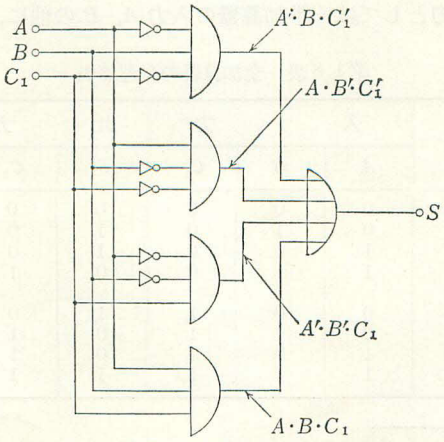
全加算器の機能を示す真理値表は，半加算器の真理値表に C_1 の条件を加えたものであり，**第1.8表**に示すとおりとなる。

全加算器の回路は，この真理値表にもとづいて作成するわけであるが，それにはまず真理値表から出力である和（S），上位へのけた上げ（C_2）のための式を導き出さねばならない。出力 S についての式を導くと，第1.8表で S が1となるのは（和に1が出るのは），No. 2，No. 3，No. 5，No. 8のときであり，それぞれ，式で示すと，

$$\left.\begin{array}{ll} \text{No. 2} & S=A'\cdot B\cdot C'_1 \\ \text{No. 3} & S=A\cdot B'\cdot C'_1 \\ \text{No. 5} & S=A'\cdot B'\cdot C_1 \\ \text{No. 8} & S=A\cdot B\cdot C_1 \end{array}\right\} \qquad (1.14)$$

となる。S が1になるのは，この No. 2，No. 3，No. 5，No. 8のいずれかの場合であるので，or 条件であり，したがって，この4つの式は or 条件を示す“∨”で結べばよい。すなわち，

$$S=A'\cdot B\cdot C'_1\vee A\cdot B'\cdot C'_1\vee A'\cdot B'\cdot C_1\vee A\cdot B\cdot C_1 \qquad (1.15)$$



第1.20図　全加算器（和）の回路

となる。この式が S に1が出るための条件を示す式である。この式を and, or の図記号を使って書き表わすと，**第1.20図**のようになる。この図は全加算器のうち，和 S を得るための回路を示している。

つぎに，出力 C_2 についてみると，第1.8表で C_2 が1になるのは，No. 4, No. 6, No. 7, No. 8 の場合である。この場合も，いずれか1つの条件が成立すれば C_2 に1が出るので，この4つの条件は or 条件であり，

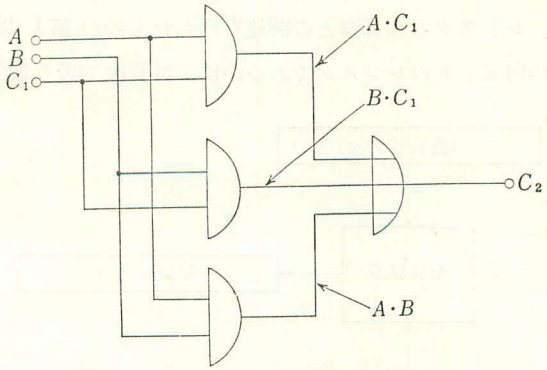
$$C_2=A\cdot B\cdot C'_1\vee A'\cdot B\cdot C_1\vee A\cdot B'\cdot C_1\vee A\cdot B\cdot C_1 \qquad (1.16)$$

となる。この式をそのまま図記号を使って書き表わせば，上位へのけた上げ C_2 を出すための回路を作ることができるが，この式は，ある手段を使ってもっと簡素化した形にすることが可能である。すなわち，簡素化した回路で等価な結果を得ることができる。

この簡素化の手段として，**論理代数（ブール代数）**[1] がある。この論理代数については，1.6で説明することとし，ここでは簡素化された式の結果だけを示すと，

$$\begin{aligned} C_2= &\ A\cdot B\cdot C_1'\vee A'\cdot B\cdot C_1\vee A\cdot B'\cdot C_1\vee A\cdot B\cdot C_1 \\ = &\ A\cdot C_1\vee B\cdot C_1\vee A\cdot B \end{aligned} \qquad (1.17)$$

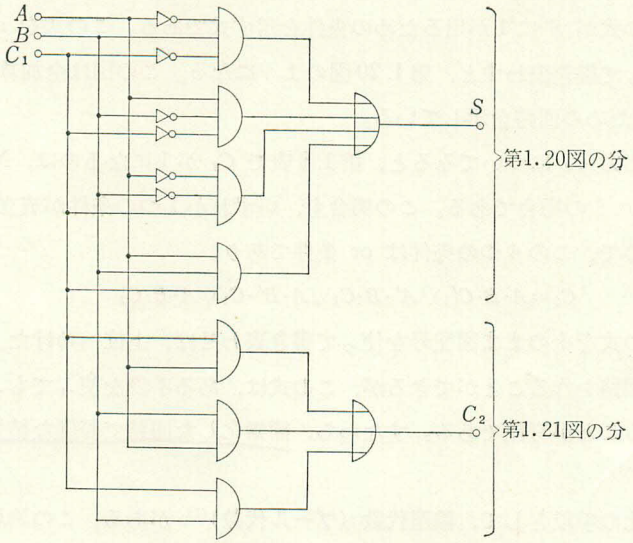
これが上位へのけた上げ条件 C_2 を示す式であり，これを and, or の図記号で表わした回路は**第1.21図**のようになる。



$C_2=A\cdot C_1\vee B\cdot C_1\vee A\cdot B$

第1.21図　全加算器（けた上げ）の回路

(1) Boolean Algebra

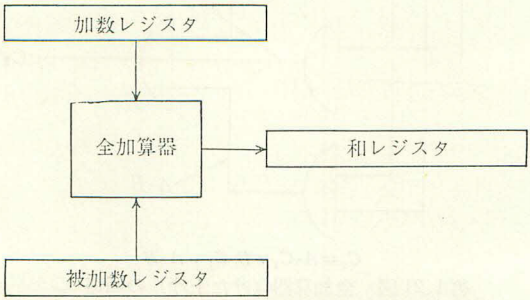


第1.22図　全加算器

これで全加算器の S（和）も，C_2（上位へのけた上げ）もできあがったわけで，この S と C_2 をいっしょにまとめてみると，**第1.22図**のようになる。

さきに，演算のための加数，被加数はそれぞれ一時記憶回路であるレジスタにたくわえられていることを述べた。したがって，加算器への入力信号はレジスタからくるわけである。レジスタと加算器との関連を示したものが**第1.23図**である。

このレジスタが何ビットのレジスタであるかは，計算機で扱う数値の大きさによ



第1.23図　加算器とレジスタとの関連

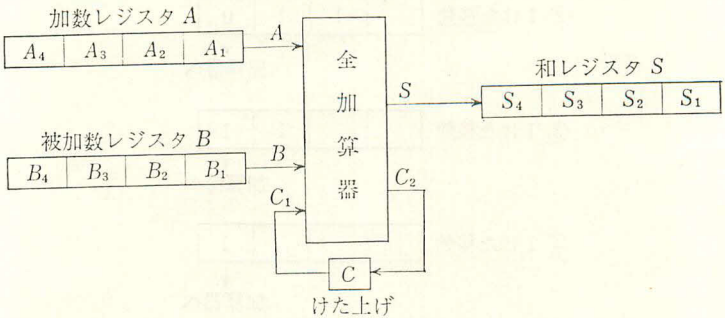
るものであるが，いま，説明を容易にするために，このレジスタを4ビットのレジスタとして話を進めることにする。

この4ビットレジスタにたくわえられている数値の加算を行なうには，2とおりの方法がある。その1つは，4ビットを並列に同時に計算する方法であり，もう1つは，この4ビットを下位のけたから順に1ビットずつ計算する方法である。

前者の4ビット同時に計算する方法を**並列加算方式**，一時に1ビットずつ加算する方法を**直列加算方式**という。

ここでは，小形計算機でよく使われている直列加算方式について説明することとする。

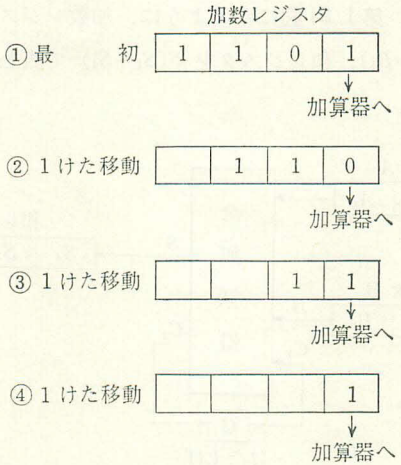
e. 直列加算器　第1.24図に示すように，加数レジスタを $A(A_1 \sim A_4)$，被加数レジスタを $B(B_1 \sim B_4)$，和レジスタを $S(S_1 \sim S_4)$ で表わし，それぞれ，4ビットのレジスタとする。



第1.24図　直列加算器

直列加算器では，加算をするための加算器は1個（1けた分）用意しておく。最初は最下位のけた $A_1 \vee B_1$ を行ない，和を S_1 に入れる。もし，そのときけた上げがあれば，けた上げ記憶部 C でそれを覚えておき，つぎのけたの演算に加える。つぎは，$A_2 \vee B_2$ を行ない，和を S_2 へ入れる。この場合にも，けた上げがあれば記憶しておく，同様に，$A_3 \vee B_3$ を S_3，$A_4 \vee B_4$ を S_4 へ入れて演算を終了する。この演算方法では，1けたの演算ごとに加算器への入力を A_1，B_1 から A_4，B_4 まで，また，出力を S_1 から S_4 まで変えてやらねばならない。このように1けたごと

に加算器への入出力の場所が変わったのではやっかいなので，入出力の場所をつねに一定にすることが考えられた。すなわち，入力はかならず最下位のけた A_1，B_1 から取り，出力はかならず最上のけた S_4 へ入れるものとする。この方法を取るためには，入力，出力レジスタにけたを移動させる機能を持たせる必要がある。たとえば，この加数レジスタに 1101 がはいっていたとすると，この加数を被加数に加えるには，まず，最下位けたの 1 を，つぎに 0 を，つぎに 1 を，そして最後に 1 をと順次加算器に送り込むのであるが，加算器への入力をかならず最下位のけたから取るものとすれば，最初の 1 を加算器に送ったあと，つぎの 0 を最下位のけたへ移動させなければならない。このけたの移動の様子を示したものが**第 1.25 図**である。



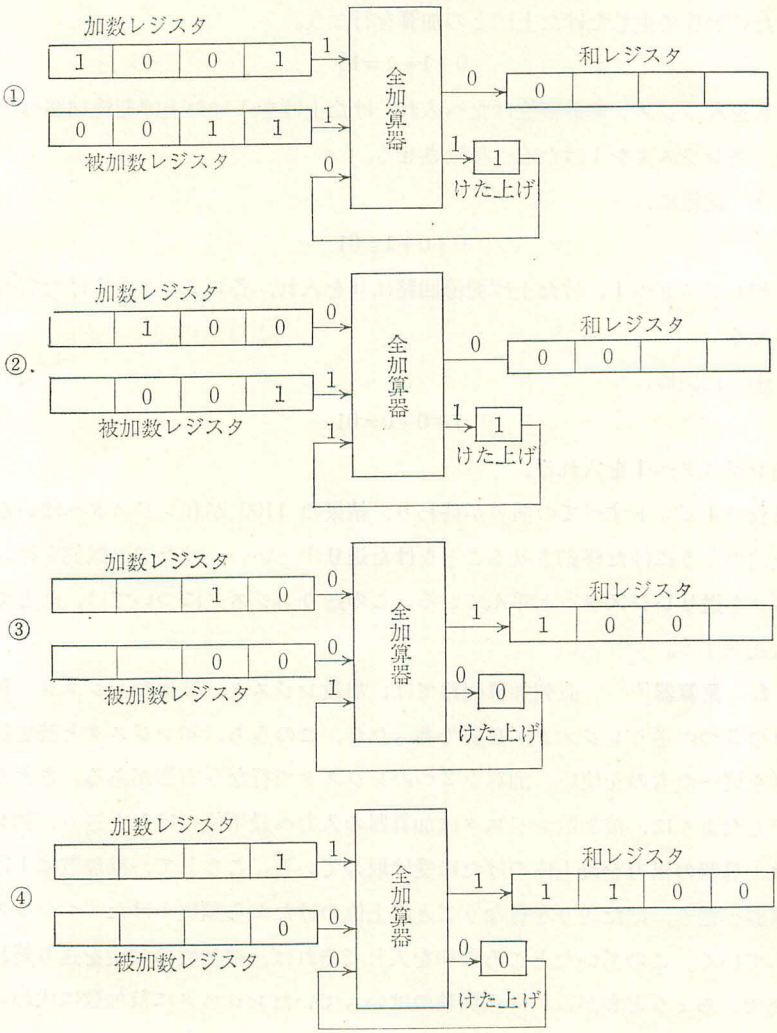
第 1.25 図　レジスタけた移動の様子

また，和レジスタの場合は，全加算器の出力をつねに最上位のけたへ入れ，順次，下位のけたへ移動させるのである。1001＋0011＝1100　の演算例における各レジスタの状態を**第 1.26 図**に示す。

加算を行なう動作はつぎのとおりである。

①　加数レジスタの最下位けたと被加数レジスタの最下位けたとを全加算器で加算する。

$$1+1=10$$



第1.26図　直列加算器の演算例

和の0を和レジスタの最高位けたへ入れ，けた上げ1をけた上げ記憶回路に記憶させる。1けたの加算が終わったので，加数レジスタ，被加数レジスタ，和レジスタの内容を1けた右へ移動させる。

② ①と同様に，加数レジスタと被加数レジスタの最下位けたと，それに前の

けたの加算で生じたけた上げとの加算を行なう。

$$0+1+1=10$$

和を和レジスタの最高位けたへ入れ，けた上げを1けた上げ記憶回路へ記憶させ，各レジスタを1けた右へ移動させる。

③　同様に，

$$0+0+1=01$$

和レジスタへ1，けた上げ記憶回路は0を入れ，各レジスタを1けた右へ移動させる。

④　同様に，

$$1+0+0=01$$

和レジスタへ1を入れる。

これで4ビットすべての演算が終わり，結果の 1100 が和レジスタへはいる。

上述のようにけた移動させることを**けた送り**(1)といい，けた送り機能を持ったレジスタを**送りレジスタ**(2)と呼んでいる。この送りレジスタについては，あとでもう一度説明する。

f.　累算器(3)　直列加算回路では，加数レジスタ，被加数レジスタ，和レジスタの3つの送りレジスタが必要であったが，このうち，和レジスタと被加数レジスタを同一のものを使い，加算を2つのレジスタで行なう方法がある。さきの例にも示したように，被加数レジスタは加算器の入力へ最下位のけたを送り，和レジスタは加算器の出力を最上位のけたに受け取っている。こうして，被加数は1けたを加算器へ送り，けた送りを行なうごとに上位のけたから順次1けたずつレジスタがあいていく。このあいたところに和を入れてやれば，被加数の数値を送り終わった時点で，ちょうど和が，前に被加数のはいっていたレジスタに被加数に代わってはいることになる。

この被加数レジスタと和レジスタを兼ねたレジスタを**累算器**と呼んでいる。累算器という言葉は，多数の加算を行なう場合，その和がつぎつぎにこのレジスタで累計されることを意味している。いま，1+2+4=7 の計算を行なう場合を考えると，

(1)　shift　(2)　shift register　(3)　accumulator

つぎのような計算動作となる。

① 累算器の全けたを0にする

② 加数レジスタへ1を入れる。

③ 加算を行なう。累算器に1がはいる。

④ 加数レジスタへ2を入れる。

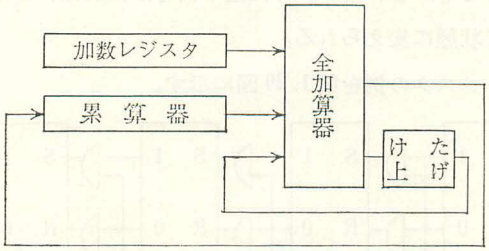
⑤ 累算器にはいっている1と加数レジスタにはいっている2の加算を行ない，同じ累算器に3がはいる。

⑥ つぎの加数4を加数レジスタへ入れる。

⑦ 累算器にはいっている3と加数レジスタにはいっている4の加算を行なう。結果の7は累算器へ入れる。

このように，演算途中の累計はかならず累算器に入れられる。

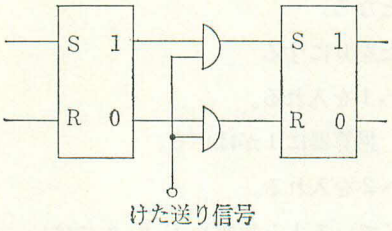
この累算器を使用した場合の加算回路は**第1.27図**のようになる。



第1.27図 累算器を使用する加算器

g. 送りレジスタ 前述のように，レジスタはフリップフロップの集まりである。したがって，送る（シフトする）ということは，1つのフリップフロップの状態（セット状態またはリセット状態）を他のフリップフロップに移し与えることを意味する。けた送りを行なうための基本回路を**第1.28図**に示す。

図の**けた送り信号**は，送りを行ないたい時点で印加するパルス信号である。いま，左側のフリップフロップがセットされているとすると，出力端子1側には出力が出ており，0側には出力がない。このときに，けた送り信号を加えてやると，上のゲートは2つの入力とも1となるので and 条件を満足し，出力に信号“1”が出て，右側のフリップフロップをセット状態にする。一方，下側のゲートにもけた

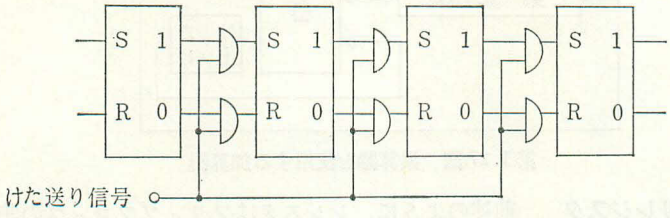


第1.28図　送りレジスタ基本回路

送り信号が加わるが，左側のフリップフロップから出力がきてないので and がとれず，このゲートから出力信号は出ない。逆に，左側のフリップフロップがリセット状態にあれば，0側に出力が出ているため，下側のゲートが開き，したがって，右側のフリップフロップがリセットされることになる。

この場合，けた送り信号が出るまでの右側のフリップフロップの状態は，セット，リセットどちらでもかまわない。けた送り信号が出れば，かならず左側のフリップフロップと同じ状態に変えられる。

4ビットの送りレジスタの例を第1.29図に示す。



第1.29図　4ビット送りレジスタ

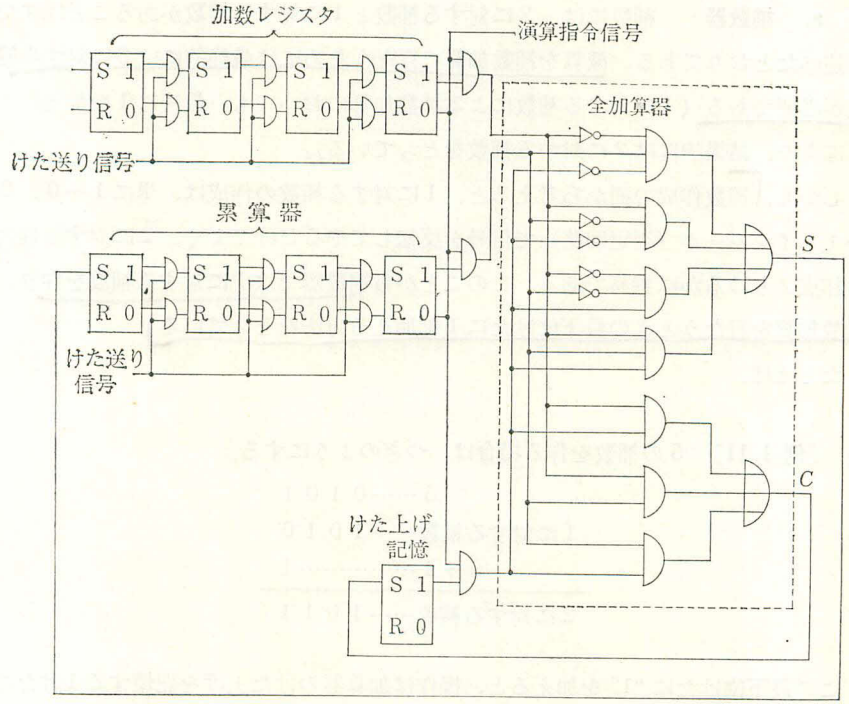
この図で，けた送り信号を与えてやれば，各レジスタ間のけた送りが同時に行なわれることになる。

注　厳密なことをいえば，この図では全フリップフロップが一番左側のフリップフロップの状態になってしまうおそれがあるが，実際の回路では，けた送り信号をフリップフロップの反転時間に比べて十分に短かくとるか，または，フリップフロップ間の信号の伝達を遅らす**遅延回路**(1)を入れて正しくけた送りされる

(1)　delay circuit

ようにしてある。

第 1.30 図は直列加算器の全体をまとめて図示したものである。図の中で，演算指令信号が全加算器の入力ゲート（and ゲート）にはいっているが，これは，演算を行なうタイミングを取るためのものであり，タイミング信号がきた瞬間に加数，被加数，けた上げのゲートが開かれ，演算が行なわれる。



第 1.30 図　直列加算器の詳細

この意味では，レジスタをけた送りさせるためのけた送り信号も，1つのタイミング信号である。ほとんどの計算機では，こうしたすべての動作がその動作の時点を指示するタイミング信号によって制御されている。

このようなタイミング信号の制御で動く計算機は，**同期式の計算機**といわれる。

特殊な計算機では，**非同期式**のものもあるが，これは計算機の動作がタイミング信号によらず，ある動作の終了がつぎの動作を誘起するといったように，いわゆる

将棋（しょうぎ）倒し式に動作する計算機である。

2. 減算回路　前述のように，計算機における減算は，補数加算により行なっている。したがって，特に減算器は必要ではなく，加算の場合の加算器がそのまま使用される。ただ，減数を補数の形にする補数器が必要であり，この補数器の出力を加算器に加えることにより，減算が実行される。

a. 補数器　補数には，2に対する補数と1に対する補数があることはすでに述べたとおりである。減算を補数加算で行なうためには最終的には2に対する補数が必要である（1に対する補数による補数加算においても，最後に1を加えることにより，結果的には2に対する補数をとっている）。

しかし，補数作成の面から考えると，1に対する補数の作成は，単に1→0，0→1とインバータ（否定回路）で信号を反転してやるだけでよく，2に対する補数の作成よりはるかに容易である。このことから補数器では1に対する補数を作り，補数加算を行なうときの最下位けたに1を加える方法をとっている。

たとえば，

〔例 1.11〕　5の補数を作る場合は，つぎのようにする。

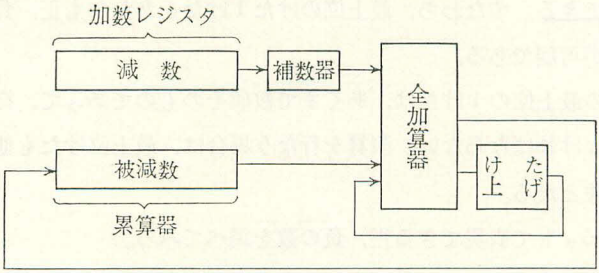
5	0 1 0 1
1に対する補数	1 0 1 0
＋1	1
2に対する補数	1 0 1 1

この最下位けたに“1”を加えると，操作は加算器のけた上げを記憶する1けたの記憶回路を使えば容易にできる。すなわち，通常の加算の場合，最下位けたの演算時には，前位よりのけた上げを示す記憶回路は“0”となっており，つぎのけたからは下位の演算結果により，“0”または“1”になる。

これに対し減算の場合には，最下位けたの演算時にこの記憶回路を“1”にしておく。その結果，最下位けたに“1”を加えたのと同じことになる。上位けたでの演算は普通の加算の場合とまったく同様である。

減算回路を**第1.31図**に示す。加算回路はそのまま使用しており，加算の場合に

加数がはいった加数レジスタは減算では減数がはいり，また，このレジスタから加算器にはいる間に**補数器**を通ることが加算の場合と異なっている。



第1.31図 減算回路

b. 負数の表現 いままでの演算例で扱った数はすべて正の数であったが，減算では結果が負の数になることも当然考えられる。そこで，負数が計算機の中でどのような表現形式になっているかについて述べることとする。

普通われわれの使う10進法では，たとえば＋3と－3のように，数値の前に＋，－の符号をつけて正負を表わしており，絶対値は同じ表現をしている。これに対し，計算機では負数は**2に対する補数**で表わされている。

〔**例 1.12**〕 10進＋3，－3の例でいえば，

＋3　　0 0 1 1

－3　　1 1 0 1（＋3の2に対する補数）

この2進法の 0011，1101 を2進加算すれば，答は当然0になる（最上位のけた上げは無視する）。すなわち，

$$\begin{array}{rr} +3 & 0\,0\,1\,1 \\ +)\ -3 & 1\,1\,0\,1 \\ \hline & \cancel{1}\,0\,0\,0\,0 \end{array}$$

いまの＋3，－3は4ビットで表示したが，これは何ビットであっても同じである。たとえば，＋5，－5を8ビットで表示すれば，

＋5　　0 0 0 0 0 1 0 1

－5　　1 1 1 1 1 0 1 1（＋5の2に対する補数）

上の＋3と－3，＋5と－5の例をみればわかるように，負の数は最上位けたが1になっている。したがって，最上位けたが1であれば負数，0であれば正数と判別することができる。すなわち，最上位のけた1けたをあたかも正，負の符号のように使うことが可能である。

しかし，この最上位の1けたは，あくまで数値そのものであって，符号ではないことに注意しなければならない。演算を行なう場合は，最上位けたも他のけたと同様に演算の対象となる。

ここで，4ビットで表現できる正，負の数を調べてみる。

第1.9表に1～8までの正，負の表わし方を示してあるが，この表から見てもわかるように，負数は－8まで表現できるが，正数は＋7までである。これを無理に＋8にしようとすると，1000 となり，－8と区別できなくなる。すなわち，＋8を表現するためには5ビットが必要となる。5ビットを用いる場合には，正数の最大数は 01111＝＋15，負の最大数は 10000＝－16 となり，かならず負の方が1だけ多く表現できることになる。

第1.9表　2進数の負数

10進数	2進正数	10進数	2進負数
＋1	0 0 0 1	－1	1 1 1 1
＋2	0 0 1 0	－2	1 1 1 0
＋3	0 0 1 1	－3	1 1 0 1
＋4	0 1 0 0	－4	1 1 0 0
＋5	0 1 0 1	－5	1 0 1 1
＋6	0 1 1 0	－6	1 0 1 0
＋7	0 1 1 1	－7	1 0 0 1
＋8	—	－8	1 0 0 0

c.　負数，正数の変換　ここでは，演算結果が負数になる場合とそうでない場合とでどのような違いがあるかを比べてみる。

〔**例 1.13**〕　①　3－5＝－2，　②　5－3＝2 をとってみると，

①	＋3	0 0 1 1	②	＋5	0 1 0 1
	－5	1 0 1 1		－3	1 1 0 1
	－2	1 1 1 0		＋2	1 0 0 1 0
		けた上げなし			けた上げあり

この例でわかるように，結果が正になる場合は最上位のけた上げが生じ，負になる場合はこのけた上げが生じない。これは減数，被減数にどのような数値をとってもいえることである。このことから，補数加算（減算）において最上位のけたからけた上げが生じないときは，結果が負数になったことを示している。

この負数を正数と同じ表現をしたいときは，負数の2に対する補数をとってやればよい（負であることをなんらかの方法で覚えておくことが必要である）。この2の補数をとるためには減算回路をつぎのように使えばよい。

① 累算器にある負数をそのまま加数レジスタに移す。

② 累算器を0にする。

③ 加数レジスタ，累算器の補数加算を行なう。

④ 結果として累算器に正数がはいる。

これを（－2）→(＋2）に変換する例で実行してみると，

まず，累算器には，－2＝1110 がはいっている。

① この 1110 を加数レジスタに移す。

加数レジスタ＝1110

② 累算器に0を入れる。

累算器＝0000

③ 加数レジスタと累算器の補数加算を行なう

加数レジスタの1に対する補数……	0 0 0 1
累算器……	0 0 0 0
＋) ＋1……	0 0 0 1
	0 0 1 0

④ 結果が累算器にはいる。

累算器＝0010

以上により，（－2）→(＋2）変換が行なわれる。

3. 乗算回路 乗算は加算回路を使用して行なうのであるが，その方法につぎの2とおりがある。

（1） 被乗数を乗数の回数だけ繰り返し加え合わせる方法，

たとえば，0110×1010 の計算は，

$$\underset{\text{(被乗数)}}{0\,1\,1\,0} \times \underset{\text{(乗数)}}{1\,0\,1\,0} = \underbrace{0\,1\,1\,0 + 0\,1\,1\,0 + \cdots\cdots + 0\,1\,1\,0}_{1\,0\,1\,0\text{回}}$$

（2）　各けたの部分積を作り，それを1けたずつ移動しながら加えていく方法（通常の筆算の方法）

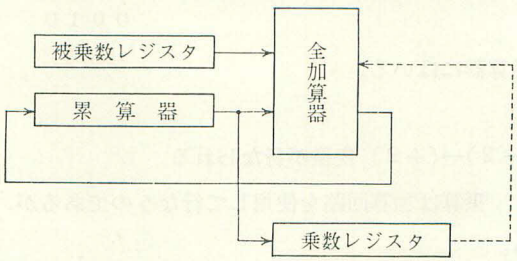
$$\begin{array}{rr} & 0\,1\,1\,0 \text{（被乗数）} \\ \times) & 1\,0\,1\,0 \text{（乗数）} \\ \hline & 0\,0\,0\,0 \\ & 0\,1\,1\,0 \\ & 0\,0\,0\,0 \\ & 0\,1\,1\,0 \end{array} \quad \text{（部分積）}$$

$$0\,1\,1\,1\,1\,0\,0 \text{（積）}$$

最初の繰り返し加算の方法は，やり方は単純であるが被乗数が大きくなると計算時間が長くなるので，一般に2番目のけた移動の方法が用いられている。

以下，後者の方法について演算回路を説明する。

この方法の回路動作は，部分積の加算とけた送りの繰り返しである。このうち，部分積については，2進演算の項で触れたように，乗数が1のときは被乗数がそのまま部分積（上の例では 0110）になり，乗数が0のときは部分積は0になっている。このことは，乗数が1のときは被乗数をそのまま前の結果に加算したのち，その結果を1けた右にけた送りさせ，乗数が0のときは加算を行なわず，前の結果をそのまま右に1けたけた送りしてやるだけでよいことになる。この手順を行なわせ



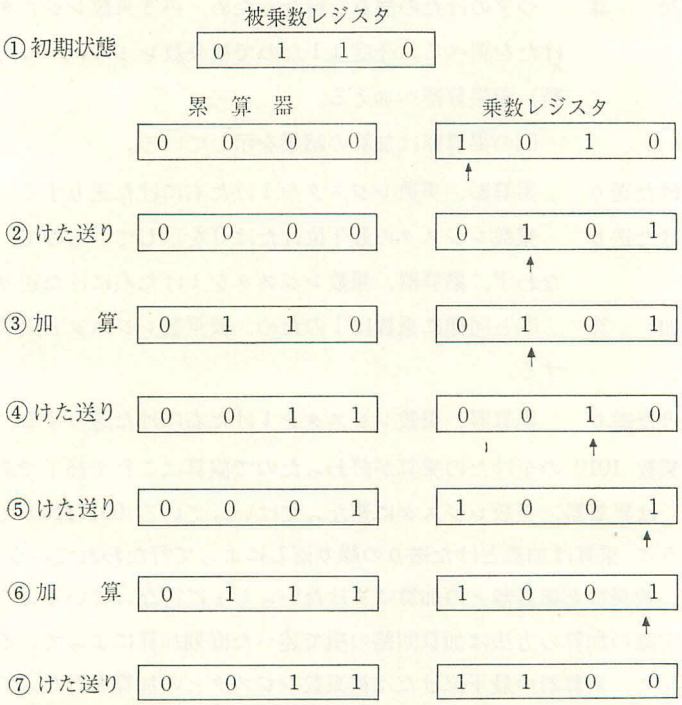
第1.32図　乗算回路

る乗算回路を**第1.32図**に示す。

この回路に示されている被乗数レジスタは，加算の場合の加数レジスタとして使用されたレジスタであり，全加算器と累算器も同一のものである。加算の場合と異なるのは，乗数を入れるための乗数レジスタが必要である。この乗数レジスタは累算器の最下位けたに続いて接続されたようになっており，最終的には，乗算結果（積）が累算器と乗数レジスタにまたがってはいることになる。

この計算動作を具体的に各レジスタに値を入れて実行してみる。

数値は前の例 0110×0101=111100 を用いることにする。まず最初に，被乗数（0110）が被乗数レジスタへ，乗数（1010）が乗数レジスタにはいったところから始める。**第1.33図**に対応して，



↑印は乗数の最上位けた位置を示す。

第1.33図 0110×1010=00111100 の演算の例

① 初期状態　被乗数，乗数がそれぞれのレジスタにはいった状態を示している。

② けた送り　最下位けたの乗算を行なうため，乗数レジスタの最下位けたを調べると0である。したがって，被乗数（部分積）の加算は行なわず，累算器，乗数レジスタを1けた右にけた送りする。この場合，累算器の最下位けたは乗数レジスタの最上位けたへ移動させる。また，乗数レジスタの最下位けたは役目が終わったのでこれを捨て，累算器の最上位けたは0を入れる。

なお，被乗数レジスタはそのままである。

これで1けたの演算が終わったことになる。

③ 加　算　つぎのけたの演算を行なうため，再度乗数レジスタの最下位けたを調べる。今度は1なので被乗数レジスタの内容（部分積）を累算器へ加える。

図の累算器は加算の結果を示している。

④ けた送り　累算器，乗数レジスタを1けた右にけた送りする

⑤ けた送り　乗数レジスタの最下位けたは0を示しているので，加算は行なわず，累算器，乗数レジスタを1けた右にけた送りする。

⑥ 加　算　③と同様に乗数は1のため，被乗数レジスタと累算器を加算する。

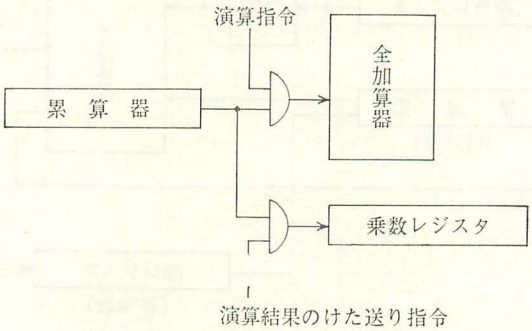
⑦ けた送り　累算器，乗数レジスタを1けた右にけた送りする。

以上で乗数 1010 の全けたの乗算が終わったので演算はこれで終了である。演算結果（積）は累算器，乗数レジスタに連なってはいっている 00111100 である。

このように，乗算は加算とけた送りの繰り返しによって行なわれている。なお，演算例では，被乗数と累算器との加算は4けたいっしょに行なっているようになっているが，実際の加算の方法は加算回路の項で述べた直列加算によっている。

したがって，累算器の最下位けたは被乗数レジスタとの加算を行なっているうちは全加算器への入力信号となり，加算が終わって1けた右へけた送りされる場合は，乗数レジスタの最上位けたへの入力となっている。この累算器出力の切り換え

を and ゲートを使って示すと，第1.34図のようになる。



第1.34図　全加算器と乗数レジスタの切り換え

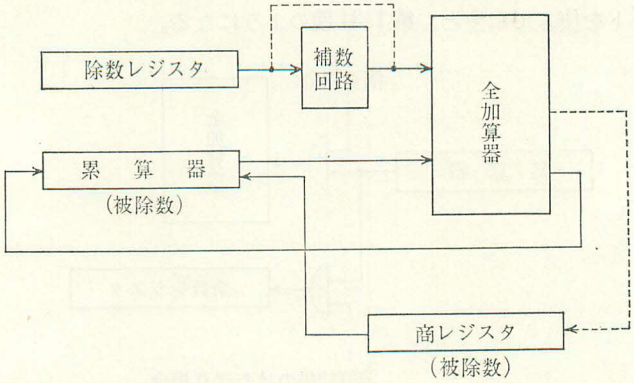
この図において，累算器と被乗数レジスタの内容を直列加算する場合は演算指令信号が出ているので，全加算器への and ゲートが開き，累算器の出力は全加算器への入力となる。この演算が終わって演算結果を1けた右にけた送りする場合は，演算結果けた送り指令が出て，累算器の出力は乗数レジスタ最高位けたへの入力となる。

このように，and ゲートはある信号を通すか通さないかの制御回路としてもよく用いられる。

4. 除算回路　乗算が加算の繰り返しによって行なわれたのに対し，除算は減算を基本として行なわれる。そしてこの除算の行ない方にも被除数から除数が何回引けるか，単純に減算を繰り返していく方法と，筆算の場合と同じように1けたずつけた送りしながら減算を行なっていく方法がある。除算の場合にもけた送りによる方法の方が減算回数が少なくてすみ，演算時間が短いので一般にこの方法がとられている。

ここでは，けた送りによる方法を説明する。この場合の演算回路は，第1.35図に示すものである。

乗算では積が累算器と乗数レジスタに連続してはいったが，除算では逆に被除数を累算器と商レジスタ（乗算の場合の乗数レジスタ）にまたがって入れておく。商は商レジスタに現われ，余りは累算器に残る。



第1.35図　除　算　回　路

また，被除数−除数の減算は，普通の減算回路と同様に補数加算によるものであり，したがって，除数は補数回路を通って全加算器へと導かれている。

この回路での除算の行ない方を述べる前に，筆算での除算を試みてみる。

〔**例 1.14**〕　例として，100010÷1100 をとる。

```
          0 0 0 0 1……商
      ┌──────────
1 1 0 0 | 1 0 0 0 1 0
          1 1 0 0
        ──────────
            1 0 1 0…余り
```

筆算で除算を行なう場合，被除数の高位のけたから，順次，除数と比較し，除数の方が大きければ商に0を立ててつぎのけたに移り，また，比較を行なう。あるけたにおいて除数の方が小さくなれば商に1を立て，除数の減算を行なって差をとり，つぎのけたに移る。

人間が筆算を行なう場合は，被除数，除数の大小判定を行ない，被除数の大きい場合にのみ，被除数−除数の減算を行なっているが，計算機では，すべてのけたについてまず減算を行ない，その結果が正のときは商に1を立て，結果が負のときは商に0を立てるやり方をとる。なお，この結果が負であるときは引き過ぎてしまっているので，再び除数を被除数に加えてもとにもどしてからつぎのけたへと進むよう

にしなければならない。

この減算結果が負であるか否かの判定は，減算回路の項で述べた補数加算で，最高位のけたからけた上げがあるかどうかを判別して行なう。たとえば，3－6の減算では，

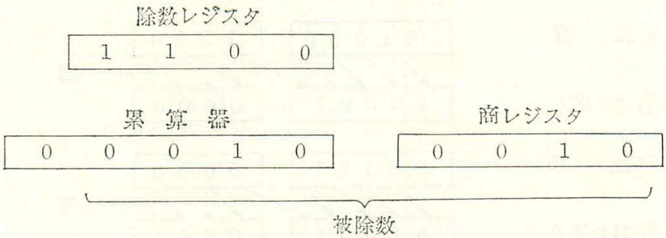
〔例 1.15〕

```
          0011     3
+)        1010     6の2に対する補数
─────────────
       (0)1101
        けた上げなし
```

この例でもわかるように，減算結果が負の場合，最高位けたからのけた上げは生じない。

以上の方法を各レジスタに数値を入れて具体的に実行してみる。数値は，筆算で行なった 100010÷1100 を使用する。

各レジスタの初期状態は，第 1.36 図に示すとおりである。

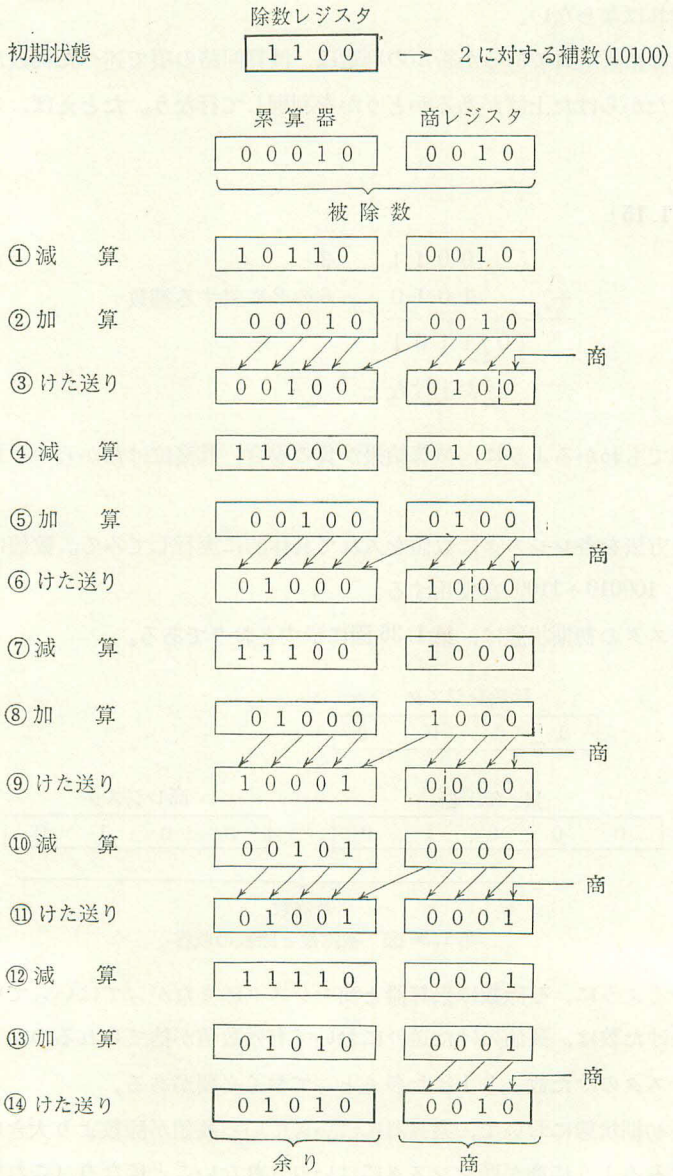


第 1.36 図　被除数と除数の収容

図にみるように，被除数は累算器と商レジスタにまたがってはいっている。なお累算器のけた数は，数値のけた送りにおいて有効数値が捨てられるのを防ぐため，除数レジスタのけた数より1けた多くとっておく必要がある。

また，初期状態において，累算器にはいっている数値が除数より大きい場合は，下の例にみるように商が商レジスタにはいりきれないことになり（これを**あふれ**[(1)]

(1)　overflow


除数レジスタ
初期状態
1 1 0 0
→ 2に対する補数(10100)
累算器
商レジスタ
0 0 0 1 0
0 0 1 0
被除数
①減　算
1 0 1 1 0
0 0 1 0
②加　算
0 0 0 1 0
0 0 1 0
③けた送り
0 0 1 0 0
0 1 0 0
商
④減　算
1 1 0 0 0
0 1 0 0
⑤加　算
0 0 1 0 0
0 1 0 0
⑥けた送り
0 1 0 0 0
1 0 0 0
商
⑦減　算
1 1 1 0 0
1 0 0 0
⑧加　算
0 1 0 0 0
1 0 0 0
⑨けた送り
1 0 0 0 1
0 0 0 0
商
⑩減　算
0 0 1 0 1
0 0 0 0
⑪けた送り
0 1 0 0 1
0 0 0 1
商
⑫減　算
1 1 1 1 0
0 0 0 1
⑬加　算
0 1 0 1 0
0 0 0 1
⑭けた送り
0 1 0 1 0
0 0 1 0
商
余り
商

第1.37図　除　算　の　経　過

という)，正しい結果は得られない。

〔例 1.16〕 11000000÷0111

```
              1 1 0 1 1……商
0 1 1 1 | 1 1 0 0 0 0 0 0
          0 1 1 1
          ---------------
            1 0 1 0
            0 1 1 1
          ---------------
              1 1 0 0
              0 1 1 1
          ---------------
                1 0 1 0
                0 1 1 1
          ---------------
                    1 1……余り
```

[ ] 内の 1100 が累算器にはいっている数値であり，除数レジスタの 0111 より大きい。この場合の商は5けたとなり，4けたの商レジスタにはいりきれなくなる。

第1.37図において，演算経過を見ると，つぎのとおりである。

① 減算： 被除数（累算器）—除数（除数レジスタ）の減算を行なう。ただし，被除数（累算器）とは，被除数のうち累算器にはいっている部分（ここでは 00010）をさす。

減算はもちろん除数を補数器を通して加算する補数加算により行なわれる。

```
        0 0 0 1 0……被除数(累算器)
+)      1 0 1 0 0……除数2に対する補数
-------------------
    (0) 1 0 1 1 0
     けた上げなし
```

減算結果は最高位からのけた上げがなく，したがって，負数である。

② 加算：減算結果が負数であったので，被除数（累算器）をもとの数にもどすための加算を行なう。この加算の場合は除数は補数器を通さず，直接，全加算器へ導びかれる（第1.35図で補数器の上の点線はこれを意味している）。

```
  1 0 1 1 0    被除数（累算器）
+)  1 1 0 0    除数
-------------
1 0 0 0 1 0    もとの被除数（累算器）
```

③　けた送り：被除数を左へ1けた送る。

このとき，商レジスタの最下位けたに，減算結果が負であったこと，すなわち，商に1が立たなかったことを示す0を入れる。

いままでの演算回路においては，各レジスタのけた送りは右送りのみであった。しかし，除算回路では左送りを使うので，累算器と商レジスタには左送り機能も持たせておく必要がある。

④　減算：①と同様の減算を行なう。

```
    0 0 1 0 0
+) 1 0 1 0 0
-------------
(0) 1 1 0 0 0
```

けた上げなし

けた上げなく負数を示している。

⑤加算：②と同様被除数の復元を行なう。

```
    1 1 0 0 0
+)    1 1 0 0
-------------
1 0 0 1 0 0
```

⑥　けた送り：③と同様のけた送りを行なう。この場合も商は0であり，商レジスタの最下位けたに0を入れる。

⑦　減算：①と同様の減算を行なう。

```
    0 1 0 0 0
+) 1 0 1 0 0
-------------
(0) 1 1 1 0 0
```

けた上げなし

⑧　加算：②と同様の復元のための加算を行なう。

```
  11100
+)  1100
------
 101000
```

⑨ けた送り：②と同様のけた送りを行なう。商は0。

⑩ 減算：①と同様に行なう。

```
  10001
+) 10100
--------
(1) 00101
```

けた上げあり

この場合は，けた上げがあるので結果は正であり，減算ができること，すなわち，商に1が立つことを意味している。

今度はもとにもどすための加算は行なわれない。

⑪けた送り：③と同様にけた送りを行なう。⑩の減算結果が正であったので商に1を入れる。

⑫ 減算：①と同様に行なう。

```
  01010
+) 10100
--------
(0) 11110
```

けた上げなし

⑬ 加算：②と同様に復元のため加算を行なう。

```
  11110
+)  1100
------
 101010
```

これまでで，③，⑥，⑨，⑪と4回のけた送りを行ない，したがって，被除数全けたについての減算を終了したことになる。

⑭ けた送り：この最後のけた送りは，商の最下位を商レジスタに入れるためのものであり，したがって，累算器のけた送りは行なわない。

この商レジスタのけた送りにおいて，最初の③のけた送りで入れた商（0）は商レジスタからはみ出すことになる。もし，このはみ出す商が1であれば，最終

結果の商が商レジスタ（この場合4けた）を越え，はじめに述べたあふれの状態となる。初期状態において除数が累算器の被除数より大きければ，このあふれは起こらない。

①～⑭の処理で除算は完全に終了し，商が商レジスタに，余りは累算器に記憶されている。

以上で，加，減，乗，除の演算回路を終わるが，どの演算もすべて全加算器を使って2の加算が演算の主体を占めており，しかも，各レジスタはすべて2の演算に共通に使われている。

各演算におけるレジスタの対応を整理して示すと，**第1.10表**のようになる。

この表では加算回路で加数のはいるレジスタをAレジスタ，乗算回路での乗数がはいるレジスタをBレジスタとして示してある。

第1.10表　レジスタの対応

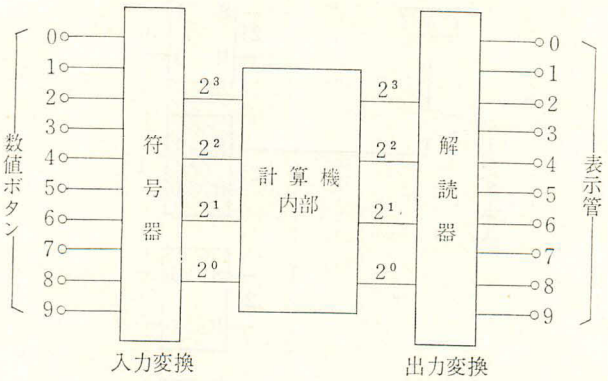
	Aレジスタ	累算器	Bレジスタ	補数器
加算回路	加数	被加数および和	使用せず	不要
減算回路	減数	被減数および差	使用せず	要
乗算回路	被乗数	積	乗数および積	不要
除算回路	除数	被除数および余り	被除数および商	要

1.5.3　入出力変換回路

計算機内部での数値の表現は，前述のように計算機の動作につごうのよい2進法で行なわれている。これに対し，われわれが計算機に数値を入れるときは，0～9の10進表示のボタンによって行ない，また，演算結果も表示窓に0～9の数字を用いて表示している。このため，入力，出力時においては，10進法↔2進法の相互変換が必要となる。

入力時において，数字ボタンによる0～9の入力を計算機の**内部コード**（2進法）に変換する回路を**符号器**(1)，逆に内部コードを0～9の表示管に変換する回路を**解読器**(2)といっている。

（1）encoder　（2）decoder



第1.38図　符号器と解読器

符号器，解読器と計算内部コードの関係は**第1.38図**に示すとおりである。

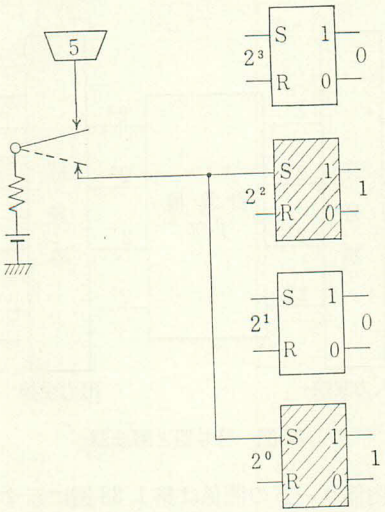
1.　入力変換回路（符号器）　入力変換回路は，0～9の数値ボタンに対応して2進信号を発生させる10進→2進変換回路である。

この変換回路を実現するのはきわめて簡単で，各数値ボタンに1つずつの接点をもうけることにより，容易に対応する2進信号を発生させることができる。

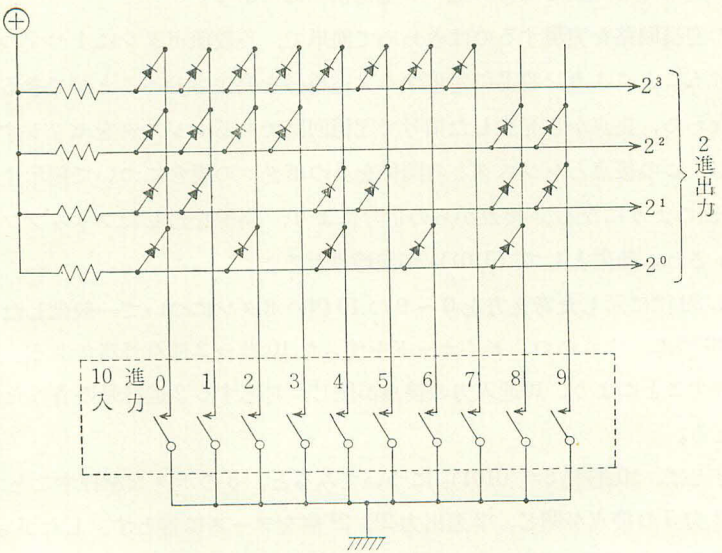
すなわち，接点から発生した信号で記憶回路であるレジスタをセットするわけであるが，この接点とレジスタとの関係を5のボタンの場合について例示すると，**第1.39図**のようになる。接点からの信号により，斜線を付したフリップフロップがセットされ，結果として $(0101)_2$ が記憶される。

第1.39図に示した考え方を0～9の10個のボタンについて一般化したものが，第1.40図に示すもので，ダイオードを使った10進→2進符号器である。数値ボタンを押すことにより，10進入力の接点が閉じ，対応する2進符号の各けたに正電圧を与える。

たとえば，10進数 $5=(0101)_2$ についてみると，5のボタンを押すことにより，10進入力5の接点が閉じ，2進出力 2^1，2^3 線をアースに落とす。したがって，2^0，2^2 線のみに ⊕ 信号が現われ，結果として0101を表示することになる。この2進出力に現われた信号により記憶回路（レジスタ）をセットしてやればよい。



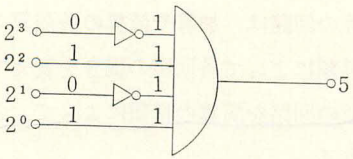
第1.39図 5のセット



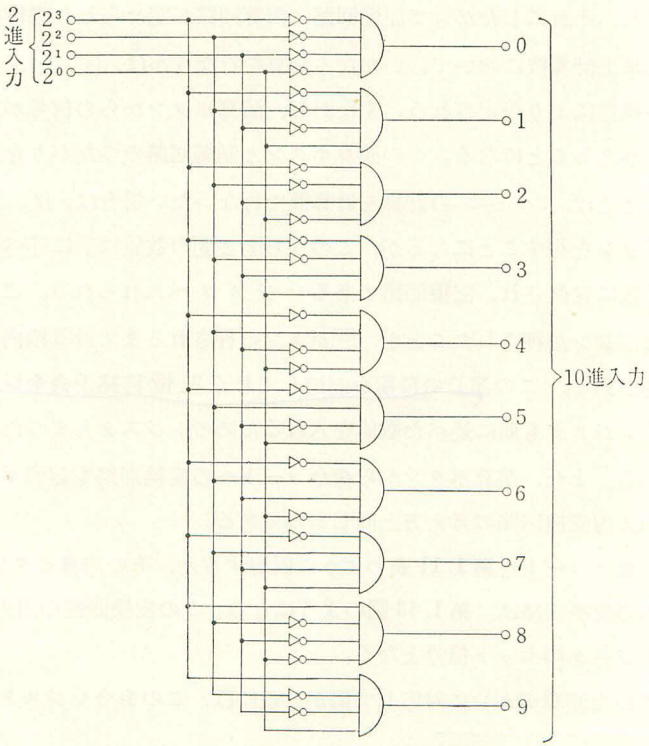
第1.40図 10進-2進符号器

2. 出力変換回路（解読器）　計算機の内部コードで，ある2進数を表示窓の10進数表示に変換する解読器は，符号器と同様にダイオードを使用しても作成できるが，一般には，and ゲートを用いて実現している。and ゲートの入力は，2^3, 2^2, 2^1, 2^0 を示す信号線であり，出力は1個の10進信号である。

たとえば，2進数 0101 を解読するための and ゲートは，**第1.41図**に示すとお



第1.41図　5の解読器



第1.42図　2進—10進解読器

りで，2^3，2^2，2^1，2^0 の信号線にそれぞれ 0，1，0，1 がきたとき and ゲートが開き，10進数5に出力が現われる。and ゲートを用いた 0～9 の全出力の解読器は，**第1.42図**のとおりである。

この 0～9 の 10 進出力が各表示窓の中の対応する数字を点燈することになる。

1.5.4　制御回路(1)

記憶回路，演算回路等の回路は，加算や減算の各演算ごとに別々に用意されているのではなく，演算の種類によって各回路の働きを変えて希望する演算を行なわせている。このように共通の回路を演算の種類によって，どのように使うかの制御を行なうのが**制御回路**である。

この制御回路からは，あるときは演算指令信号が，またあるときは，けた送り信号が出され，それにしたがって記憶回路，演算回路が定められた動作を行なう。

電子式卓上計算機において，いかなる演算を行なうかは，[+]，[−]，[×]，[÷]の演算ボタンの操作により決定される。すなわち，演算ボタンからの信号が制御回路に対し命令を与えることになる。この演算ボタンと制御回路のつながりを調べてみる。

いまたとえば，2+3=5 の計算を計算機で行ないたい場合は，[2]，[+]，[3]，[=]の順序でボタンを押すことになるが，このうち[2]と[3]の数値は前に述べた入力変換回路で2進数に変換され，記憶回路であるレジスタへ入れられる。これと同じように，[+]のボタンが押されたことを，[=]ボタンの押されるまで計算機内部で記憶しておく必要がある。この演算の種類を記憶しておく記憶回路を**命令レジスタ**(2)という。このレジスタも前に述べた数値を入れるためのレジスタとまつたく同じ構成になっている。また，演算ボタンから命令コードへの変換回路も数値ボタンからレジスタへの入力変換回路の考え方と同じものである。

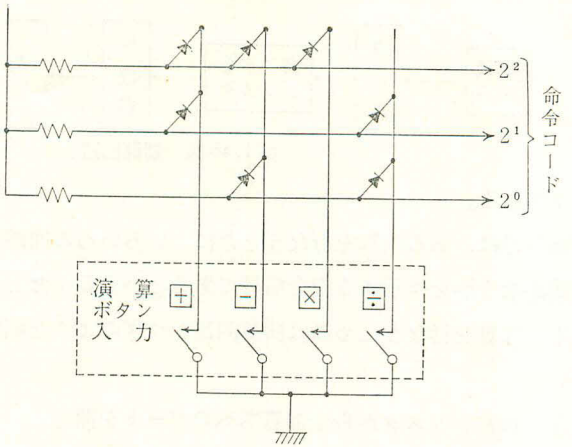
いま，命令コードを**第1.11表**のように仮定すると，その演算ボタンから命令レジスタへの変換回路は，**第1.43図**のようになる。この変換回路の出力信号(2^0～2^2)が命令レジスタのセット信号となる。

制御回路を演算ボタンに対応して働かせるには，この命令レジスタを解読し，そ

(1)　control circuit　　(2)　instruction register, control register

第1.11表
命令コード表

演算の種類	命令コード
加　算	001
減　算	010
乗　算	011
除　算	100

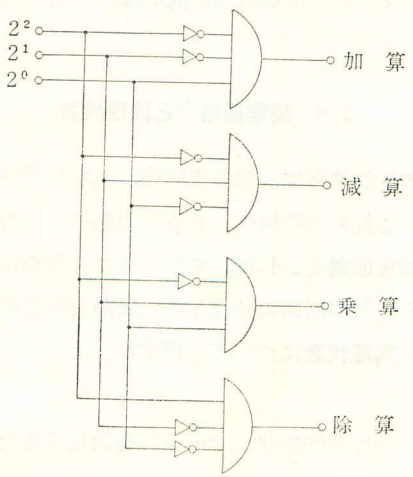


第1.43図　命令コード符号器

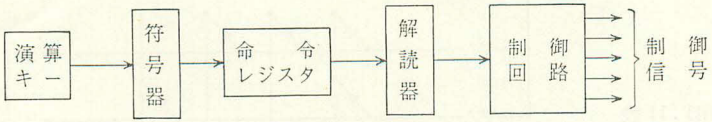
の解読した信号を制御回路に与えてやらねばならない。この解読の方法には，さきの2進→10進の出力変換で述べたandゲートによる解読器と同じ方法がとられる。

命令コードの解読器を**第1.44図**に示す。

この命令コード解読器からの出力が制御回路にはいり，そこではじめて制御回路から記憶回路や演算回路への制御信号が出される。この関係は，**第1.45図**に示す



第1.44図　命令コード解読器



第1.45図　制御回路

とおりである。

制御信号は，ある演算を行なうときに，いろいろの回路を，いつ，どのような順序で働かせるかを指示する指令信号であるといえる。たとえば，加算を例にとってみても，加算を行なうためには演算回路をつぎのような順序で動作させなければならない。

① 加数レジスタから全加算器へのゲートを開く。

② 累算器から全加算器へのゲートを開く。

③ 全加算器を働かせる。

④ 加数レジスタを右へ1けたけた送りさせる。

⑤ 累算器を右へ1けたけた送りさせる。

⑥ 全加算器からの出力（和）を累算器へ入れる。

このような各回路の動作順序，動作タイミングを決めているのが制御信号である。この制御信号を出すための制御回路の構成は，かなり複雑になるので，ここでは省略する。

1.6　論理回路(1)と論理代数

演算回路の中心である加算器は，論理積回路，論理和回路，否定回路の組み合わせで構成されている。これらの回路は，ある入力条件と出力の**論理**（ロジック）を示す回路であるから**論理回路**といわれ，また，その論理を示す式を**論理式**と呼ばれている。ここでは，その論理回路の整理と論理回路を効果的に働かせるための手段として用いられている**論理代数**について説明する。

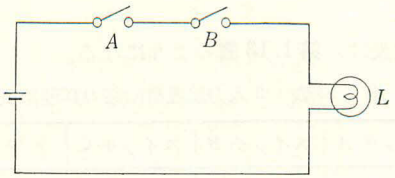
1.6.1　論理回路

1.　論理積回路　論理積回路は，and 条件を満足するためのものであり，その

(1)　logic circuit

基本的な考え方は加算回路の項で述べたとおりであるが，ここでは，そのand条件をもつ論理積回路をランプを点火するスイッチ回路の例で説明する。

第1.46図において，スイッチA，スイッチBは通常開いているものとすると，



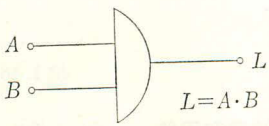
第1.46図　論理積回路

ランプLが点火するのは，スイッチA，スイッチBがともに動作した場合のみである。

スイッチA，スイッチBは動作して閉じるか，動作しないで開いているかの2つの状態のいずれかをとり，またランプLも，点火するか滅火するかの2つの状態のいずれかを取る。したがって，スイッチが動作している状態を1，動作していない状態を0，また，ランプLの点火状態を1，滅火状態を0に対応させてスイッチA，BとランプLの関係を表に示すと**第1.12表**のようになる。

第1.12表　論理積回路の真理値表

スイッチA	スイッチB	ランプL
0	0	0
0	1	0
1	0	0
1	1	1



第1.47図　論理積回路の図記号

この表は，スイッチA，BとランプLの組み合わせを示す真理値表であり，スイッチA，Bは，ランプLのand条件であることを示している。すなわち，

$$L=A\cdot B \tag{1.18}$$

と表わされる。

また，これを図記号で表わすと，**第1.47図**のようになる。この場合，入力，出力という言葉を使うと，入力はスイッチA，Bであり，出力はランプLである。

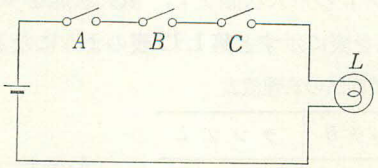
これまでのand条件は2入力のもののみを示してきたが，入力はいくつあって

もよく，たとえば3入力の and 条件をもつランプ回路を示すと，**第1.48図**のようにスイッチ A，B，C を直列に接続したものとなる。この回路でランプ L を点火させるためにはスイッチ A，B，C を同時に動作させて閉じてやらなければならない。

この3入力の真理値表は，**第1.13表**のようになる。

第1.13表　3入力論理積回路の真理値表

スイッチ A	スイッチ B	スイッチ C	ランプ L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



第1.48図　3入力の論理積回路

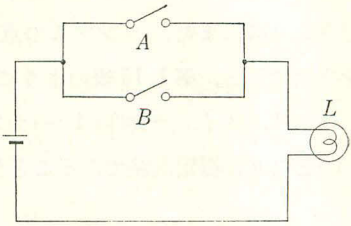
2.　論理和回路　　or 条件をもつ論理和回路を and 条件と同様に，ランプ点火回路の例で示すと，**第1.49図**に示すように，スイッチ A，スイッチ B を並列に接続したものとなる。

この回路において，ランプ L を点火させるにはスイッチ A，スイッチ B のいずれかが動作すればよい（もちろん両方動作してもランプは点火する）。このことは，スイッチ A，スイッチ B はランプ L を点火するための or 条件になっていることを示している。

この場合も，スイッチ A，B の動作を1，不動作を0に，ランプ L の点火を1，滅火を0に対応させて真理値表を示すと**第1.14表**のようになる。

第1.14表 論理和回路の真理値表

スイッチ A	スイッチ B	ランプ L
0	0	0
0	1	1
1	0	1
1	1	1



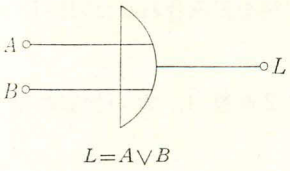
第1.49図 論理和回路

この論理和回路を論理式で表わすと，

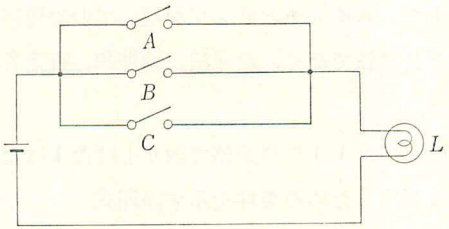
$$L=A\vee B \tag{1.19}$$

となり，その図記号表示は**第1.50図**である。

また，論理和回路の入力（スイッチ）もいくつあってもよく，3入力の論理和回路を示すと，**第1.51図**のとおりである。

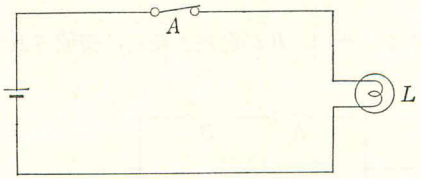


第1.50図 論理和回路の図記号



第1.51図 3入力の論理和回路

3. 否定回路 否定回路もランプ点火回路で示すと，**第1.52図**のようになる。



第1.52図 否定回路

論理積回路，論理和回路でのスイッチは，通常動作していないときに開いた状態になっており，動作することにより回路を閉じたが，否定回路では，逆にふだんはスイッチが閉じており，動作することによりスイッチを開くように設定してある。このようにしておけば，スイッチ A 不動作（閉）のときはランプ L が点火し，スイッ

チ A が動作すると滅火することになる。この場合も，スィッチ A が動作を1，不動作を0に，また，ランプ L の点火を1，滅火を0に対応させて，その真理値表を作成すると，**第1.15表**のようになる。

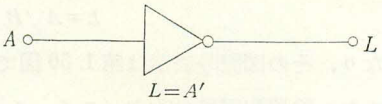
これは，スイッチ動作(1)→ランプ滅火(0)，スイッチ不動作(0)→ランプ点火(1)となり，否定回路であることを示している。この否定回路の論理式は，

$$L=A' \tag{1.20}$$

となる。また，図記号表示は，**第1.53図**のとおりである。

第1.15表　否定回路の理真値表

スイッチ A	ランプ L
0	1
1	0



第1.53図　否定回路の図記号

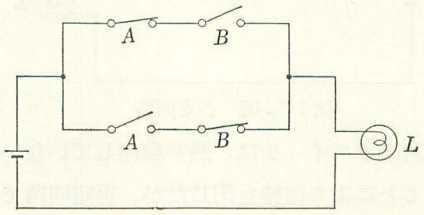
以上で，スイッチとランプを用いての論理回路を示したが，これらの論理回路はそれぞれ単独でなく，論理積，論理和，否定各回路を組み合わせた回路も作成可能である。

例として，1.4加算回路で取り上げた1けたの2進数 A，B の加算の和 (S) に信号1が出るための条件を示す論理式

$$S=A'\cdot B\vee A\cdot B' \tag{1.21}$$

を満足する回路を，いままでに説明してきたスイッチとランプで表わすと，**第1.54図**のようになる。

この図において，スイッチ A，B が直列と並列に接続されているが，上段の直列



$L=A\cdot B\vee A'\cdot B'$

第1.54図　和を示す回路

接続スイッチ A, B は $A'\cdot B$ を示し，下段は $A\cdot B'$ を表わしている。論理式

$$S=A'\cdot B\vee A\cdot B' \tag{1.22}$$

は $A'\cdot B$ と $A\cdot B'$ が or 条件であることを示しているので，上段 $A'\cdot B$ と下段 $A\cdot B'$ を並列に接続すれば，この論理式を満足することになる。

したがって，この回路でランプ L が点火するのは，この論理式を満足すること，すなわち，和 (S) に1が出ることを意味している。

1.6.2 論理代数

1.5の加算回路の全加算器の項において，上位けたへのけた上げ条件を示す論理式は，

$$C_2=A\cdot C_1\vee B\cdot C_1\vee A\cdot B \tag{1.23}$$

で表わされているが，この式は全加算器の必要条件を示す真理値表から導びいた論理式

$$C_2=A\cdot B\cdot C_1'\vee A'\cdot B\cdot C_1\vee A\cdot B'\cdot C_1\vee A\cdot B\cdot C_1 \tag{1.24}$$

を簡素化したものであった。

この式 (1.23) と式 (1.24) が結果として等価なものであれば，式 (1.23) の方が回路素子が少なくてすみ，効果的であることは明らかである。

この論理式の変形，簡素化等に用いられるのが論理代数である。論理代数は，もともと，**論理学**を数学的に解析するための手段として考えられたもので，1847年に**ブール**(1)により創始されたことから，ブール代数とも呼ばれている。

論理学は，ある**命題**(2)（たとえば，「人は動物である」とか，「くじらは魚の一種である」とか）が正しい（**真**）(3)か正しくない（**偽**）(4)か，また，その命題を組み合わせたものが真であるか偽であるかを論ずるものである。この例の命題では，前者は真であり，後者は偽である。

論理代数は，この真，偽についての論理の体系を代数的に論じようというものである。

論理代数を，いまわれわれが考えているような信号の有（1），無（0），スイッチの on（1），off（0）等の2値回路に応用したのが，日本では中島章，榛沢正

(1) G. Ecole (2) proposition (3) truth (4) false

(1937年)，外国ではシャノン[1](1938年) である。

現在この論理代数は，電子計算機をはじめ，科学の広い分野において**論理設計**[2]に応用されている。

この論理代数は，普通の代数とはかなり様相を異にしているが，その基本的な規則はきわめて単純なものである。まず，この基本的な規則を考えてみる。

1.　論理積と論理和　基本的な演算は，すでにいままでに述べてきた論理積と論理和である。

a.　論理積

$$0 \cdot 0 = 0 \tag{1.25}$$

$$0 \cdot 1 = 0 \tag{1.26}$$

$$1 \cdot 0 = 0 \tag{1.27}$$

$$1 \cdot 1 = 1 \tag{1.28}$$

すなわち，2数の両方とも1の場合のみ結果は1となっている。

b.　論理和

$$0 \vee 0 = 0 \tag{1.29}$$

$$0 \vee 1 = 1 \tag{1.30}$$

$$1 \vee 0 = 1 \tag{1.31}$$

$$1 \vee 1 = 1 \tag{1.32}$$

論理和は，2数のいずれかが1であれば結果は1となっている。

この論理積と論理和は，いままで扱ってきたand条件とor条件を数字1と0を使って表わしたものにすぎない。

この基本的な演算定理から，つぎの定理が導びかれる。

$$A \vee 0 = A \tag{1.33}$$

$$A \cdot 0 = 0 \tag{1.34}$$

$$A \vee 1 = 1 \tag{1.35}$$

$$A \cdot 1 = A \tag{1.36}$$

$$A \vee A = A \tag{1.37}$$

---

(1)　C.E. Chanon　(2)　logical design

$$A \cdot A = A \tag{1.38}$$

この場合のAは，1か0の2つの値のみを取り得る変数で，論理代数でも普通の代数と同じように任意の変数を用いることができる。この論理式に用いる変数を**論理変数**といっている。

上の6つの式の証明は，論理変数Aに1，0を入れて，式(1.25)～式(1.32)を用いて論理積，論理和の演算を行なってみればよい。

式(1.33)の$A \vee 0 = A$を調べてみると，

A	$A \vee 0 = A$
0	$0 \vee 0 = 0$
1	$1 \vee 0 = 1$

結果は，$A=0$のときは，0，$A=1$のときは1となり，この式が正しいことが証明される。

式(1.34)の$A \cdot 0 = 0$も同様に，

A	$A \cdot 0 = 0$
0	$0 \cdot 0 = 0$
1	$1 \cdot 0 = 0$

となり，結果はAの1，0にかかわらず0となっている。

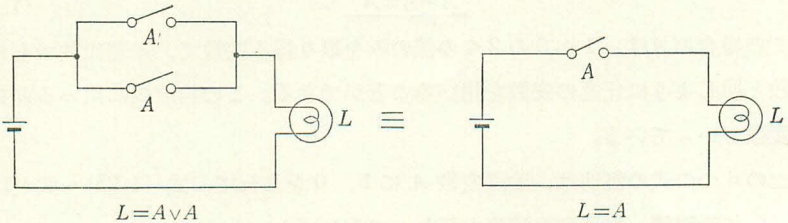
式(1.37)$A \vee A = A$についても，

A	$A \vee A = A$
0	$0 \vee 0 = 0$
1	$1 \vee 1 = 1$

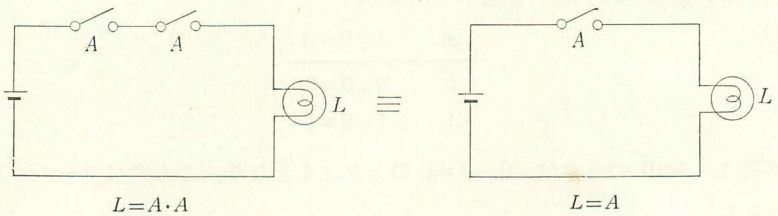
となり，$A=0$のとき結果は0，$A=1$のときは結果は1となって，式(1.37)が成立することが示される。

その他の式も同様に，Aに1または0を入れて演算を行なうことにより，正しいことが証明できる。

式(1.37)$A \vee A = A$，式(1.38)$A \cdot A = A$が成り立つことをさっきのランプ回路を用いて示したものが，**第1.55図**，**第1.56図**である。この図でAの接点がいくつあっても，いっしょに動作することを考えれば上の式が容易に理解できよう。



第1.55図



第1.56図

以上のことから，つぎのような定理を導びくことができる。

2. 変換・結合・分配の定理

a. 変換の定理

$$A \vee B = B \vee A \tag{1.39}$$

$$A \cdot B = B \cdot A \tag{1.40}$$

b. 結合の定理

$$A \vee (B \vee C) = (A \vee B) \vee C \tag{1.41}$$

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C \tag{1.42}$$

c. 分配の定理

$$A \cdot (B \vee C) = A \cdot B \vee A \cdot C \tag{1.43}$$

$$A \vee (B \cdot C) = (A \vee B)(A \vee C) \tag{1.44}$$

3. 否定の定理　　否定に関する定理には，つぎのようなものがある。

① $A'' = A$ (1.45)

② $A \vee A' = 1$ (1.46)

③ $A \cdot A' = 0$ (1.47)

式 (1.45) における A'' は，変数 A の**二重否定**を表わし，否定の否定であるから，もとの A にもどることを示している。これも，A に 1，0 を入れてみれば，すぐに証明できる。

A	A'	A''
0	1	0
1	0	1

また，式 (1.46)，式 (1.47) も A に 1，0 をあてはめてみれば証明は容易であろう。

4. ド・モルガンの定理[1]　否定に関する重要な定理に，ド・モルガンの定理がある。この定理は論理積と論理和の変換定理であり，論理式の簡素化によく使用される。

$$① \quad (A \vee B)' = A' \cdot B' \tag{1.48}$$

$$② \quad (A \cdot B)' = A' \vee B' \tag{1.49}$$

この両式は，変数の数に関係なく成立する。すなわち，

$$(A \vee B \vee C \vee D \vee E)' = A' \cdot B' \cdot C' \cdot D' \cdot E'$$

$$(A \cdot B \cdot C \cdot D \cdot E)' = A' \vee B' \vee C' \vee D' \vee E'$$

ド・モルガンの定理の証明も A，B に 1，0 を入れてみればよい。この関係は，**第 1.16 表**に示されている。

第 1.16 表　ド・モルガンの定理の証明

A	B	A'	B'	$A' \cdot B'$	$A \vee B$	$(A \vee B)'$	$A \cdot B$	$(A \cdot B)'$	$A' \vee B'$
0	0	1	1	1	0	1	0	1	1
0	1	1	0	0	1	0	0	1	1
1	0	0	1	0	1	0	0	1	1
1	1	0	0	0	1	0	1	0	0

5. その他　定理ではないが，つぎの式は知っておくと便利である。

$$A \vee A \cdot B = A \tag{1.50}$$

この式の証明は分配の定理を使えばよい。

(1) De Morgan's theorem

$$A \vee A \cdot B = A(1 \vee B)$$

$$1 \vee B = 1$$

$$\therefore \quad A \vee A \cdot B = A \cdot 1 = A$$

これまで，論理代数の定理を説明してきたが，ここで，これらの定理を用いて最初に示した全加算器の上位けた上げの論理式の簡素化を行なってみる。

真理値表から導びかれた論理式は，

$$C_2 = A \cdot B \cdot C_1' \vee A' \cdot B \cdot C_1 \vee A \cdot B' \cdot C_1 \vee A \cdot B \cdot C_1 \qquad (1.51)$$

である。これに式（1.37）の$A \vee A = A$ を応用して，$A \cdot B \cdot C_1$ を2項追加し，整理すると，

$$\begin{aligned} C_2 &= A \cdot B \cdot C_1' \vee A' \cdot B \cdot C_1 \vee A \cdot B' \cdot C_1 \vee A \cdot B \cdot C_1 \vee A \cdot B \cdot C_1 \vee A \cdot B \cdot C_1 \\ &= A \cdot B \cdot C_1' \vee A \cdot B \cdot C_1 \vee A' \cdot B \cdot C_1 \vee A \cdot B \cdot C_1 \vee A \cdot B' \cdot C_1 \vee A \cdot B \cdot C_1 \\ &= A \cdot B(C_1' \vee C_1) \vee B \cdot C_1(A' \vee A) \vee A \cdot C_1(B' \vee B) \\ &= A \cdot B \cdot 1 \vee B \cdot C_1 \cdot 1 \vee A \cdot C_1 \cdot 1 \\ &= A \cdot B \vee B \cdot C_1 \vee A \cdot C_1 \end{aligned} \qquad (1.52)$$

この式の簡素化の例をみると，機械的に式の簡素化ができる部分とそうでない部分とがある。この例では最初に行なった，

$$A \cdot B \cdot C_1 = A \cdot B \cdot C_1 \vee A \cdot B \cdot C_1 \vee A \cdot B \cdot C_1$$

と同一項を追加することにより，式の簡素化ができるのであるが，これを直観的に見出すのはかなり困難である。そこで，式の簡素化をさらに容易にするため，図を使っての簡素化を行なう方法が考えられた。こうした図にはいろいろあるが，ここではその1例として，**カルナウ図**(1)による簡素化の方法を紹介する。

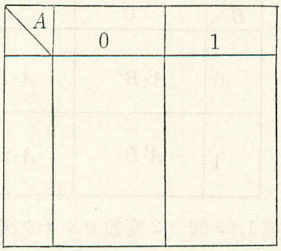
1.6.3　カルナウ図

カルナウ図は，論理代数の式を簡素化する方法として利用されている。扱う論理変数はいくつでもよいが，ここでは，4変数までのものについて説明する。

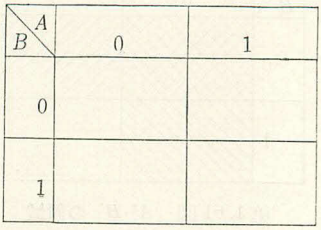
カルナウ図は，論理式を図形のうえで表現しようとするもので，論理式の領域を示すます目の集合で構成されている。このます目の数は論理変数の取り得る組み合わせの数だけ用意される。

（1）Karnaugh map

まず，1変数のカルナウ図を考えてみる。いま，変数をAとすれば，この変数Aがとり得る値はAとA'の2つのみである。したがって，このカルナウ図は，**第1.57図**に示すものとなる。この図表の左半分のます目はA'の領域を示し，右半分のます目はAの領域を示している。また，ます目上に書かれている0，1はA'，Aを表わすものである。



第1.57図　1変数のカルナウ図

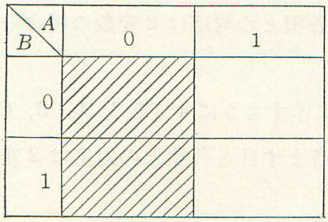


第1.58図　2変数のカルナウ図

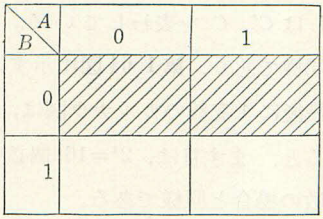
つぎに，2変数の場合を考えてみる。この2変数をA，Bとすれば，変数Aのとり得る値はA，A'，変数Bのとり得る値はB，B'であり，その組み合わせは，$A\cdot B$，$A\cdot B'$，$A'\cdot B$，$A'\cdot B'$の4通である。したがって，2変数のカルナウ図は4つのます目が必要であり，そのカルナウ図は，**第1.58図**に示すものとなる。

この図の見方を説明すると，まず，変数Aについては1変数の場合と同様に図の左半分（**第1.59図**で斜線を付した部分）はA'を表わし，右半分はAを表わしている。一方，変数Bについては，図の上半分（**第1.60図**で斜線を付した部分）はB'を表わし，下半分はBを表わすことになる。

いま，カルナウ図において，$A'\cdot B'$を表わす領域はどこかを考えてみると，$A'\cdot B'$とはA'の領域であり，かつ，B'の領域であるところ，すなわち，A'とB'

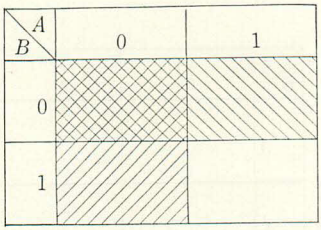


第1.59図　A'の領域



第1.60図　B'の領域

の共通の領域が$A'\cdot B'$領域である。これを**第1.61図**において，A'の領域を ▨ で，また，B'の領域を ▨ で表示してみると，$A'\cdot B'$は ▨ と ▨ の重なったところの ▩ の部分であることがわかる。他の領域についても同様に分類していくと，**第1.62図**に示すようになる。



第1.61図　$A'\cdot B'$の領域

B \ A	0	1
0	$A'\cdot B'$	$A\cdot B'$
1	$A'B$	$A\cdot B$

第1.62図　2変数カルナウ図の各領域

ではつぎに，3変数のカルナウ図を調べてみる。3変数をA，B，Cとすれば，その組み合わせは，$2^3=8$とおりとなる。したがって，ます目は8個となり，このカルナウ図は，**第1.63図**のようになる。この図の上に書かれたA，Bの00，01，11，10は，それぞれ，$A'\cdot B'$，$A'\cdot B$，$A\cdot B$，$A\cdot B'$を表示しており，Cの

C \ AB	00	01	11	10
0				
1				

第1.63図　3変数のカルナウ図

0,1はC'，Cを表わしている。各ます目と各項との対応は2変数の場合の考え方と同様であり，**第1.64図**に示すようにとなる。

最後に4変数のカルナウ図は，**第1.65図**に示すように4変数をA，B，C，Dとすると，ます目は，$2^4=16$個必要となる。各ます目と各項との対応は2変数，3変数の場合と同様である。

論理式の各項とます目との関係は以上のとおりであるが，実際にカルナウ図に論

C \ AB	0 0	0 1	1 1	1 0
0	$A'\cdot B'\cdot C'$	$A'\cdot B\cdot C'$	$A\cdot B\cdot C'$	$A\cdot B'\cdot C'$
1	$A'\cdot B'\cdot C$	$A'\cdot B\cdot C$	$A\cdot B\cdot C$	$A\cdot B'\cdot C$

第1.64図　3変数カルナウ図の各領域

CD \ AB	0 0	0 1	1 1	1 0
0 0				
0 1				
1 1				
1 0				

第1.65図　4変数カルナウ図

理式を表示する場合は，その論理式の各項に該当するます目に1を記入する。たとえば，2変数の論理式 $f=A\cdot B\vee A'\cdot B'$ をカルナウ図に表示すると，**第1.66図**のようになる。また，3変数の論理式 $f=A\cdot B\cdot C\vee A'\cdot B'\cdot C'\vee A'\cdot B\cdot C$ をカルナウ図表に表示すると，**第1.67図**のとおりである。

このようなカルナウ図を用いることにより，つぎのように論理式の簡素化を行な

B \ A	0	1
0	1	
1		1

第1.66図　$f=A\cdot B\vee A'\cdot B'$

C \ AB	00	01	11	10
0	1			
1		1	1	

第1.67図　$f=A\cdot B\cdot C\vee A'\cdot B'\cdot C'\vee A'\cdot B\cdot C$

うことができる。まず，簡単な変数の例として，

$$f=A'\cdot B'\vee A'\cdot B \tag{1.53}$$

を考えてみよう。この例をさきの論理代数の定理を使って簡素化すると，つぎのようになる。

$$\begin{aligned} f&=A'\cdot B'\vee A'\cdot B \\ &=A'(B'\vee B) \qquad (B'\vee B=1) \\ &=A' \end{aligned} \tag{1.54}$$

一方，$f=A'\cdot B'\vee A'\cdot B$ をカルナウ図に書き入れると，**第1.68図**のようになる。

B \ A	0	1
0	1	
1	1	

第1.68図　$f=A'\cdot B'\vee A'\cdot B=A'$

この図で，1を記入した部分は，さきの第1.59図で，斜線を付した部分と同一である。すなわち，A' を示している。このようにカルナウ図によれば，直観により A' を読みとることができる。この例はごく簡単なものであるので，一目で簡素化が行なえたが，もう少し複雑になると一目で簡素化というわけにはいかず，ある規則にしたがって簡素化を進める必要がある。

この規則は，つぎのようなものである。

① 論理式の各項に該当するます目に1を記入する。

② 記入された1をつぎの条件により線で囲み**ループ**を作る。

ⓐ **同一行**に隣接している偶数個の1

ⓑ **同一列**に隣接している偶数個の1

ⓒ ⓐ，ⓑにより囲まれたループが同一行または同一列に隣接して偶数個あるもの

ⓓ ます目の最上行と最下行，および，最左列と最右列は隣接しているもの

とみなす。

ⓔ 同一ます目は何回使ってもよい。

ⓕ 隣接していない1や対角線にある1は囲むことができない。

以上のことを全部について行ない，終わったらつぎの手順で簡素化した式を求める。

① 1つのループから1つの項を，変数の**論理積**の形でとり出す。

② ①の場合，1と0の両方を含む変数は**省略**することができる。

③ 各ループからとり出した項を**論理和**“∨”で結ぶ。

つぎに，これらの規則を用いて簡素化を行なう手順を実例について示す。

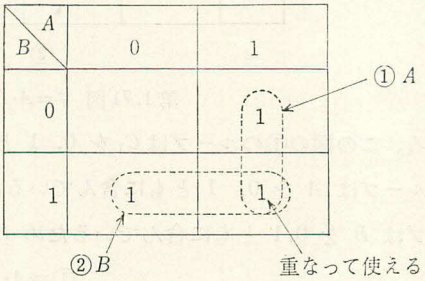
〔**例 1.17**〕

$$f=A\cdot B\vee A'\cdot B\vee A\cdot B' \tag{1.55}$$

この式をカルナウ図に示すと，**第1.69図**となる。この図の1をさきの簡素化の手順にしたがって線で囲みループを作ると，**第1.70図**に示すような2つの

A / B	0	1
0		1
1	1	1

第1.69図 $f=A\cdot B\vee A'\cdot B\vee A\cdot B'$



第1.70図 $f=A\vee B$

ループができる。このうち，①のループは同一列に1が2個(偶数)連なっており，②のループは同一行に1が2個連なっている。ここで，$A\cdot B$ を示す右下の1は両方のループに共通に使われている。この2つのループから論理式を導びき出すのであるが，まず①のループをみると，これは B を0，1とも含んでおり，さきの手順で示した0，1の両方をもつ変数は省略することができるということから，B を省略し，①のループ$=A$ となる。一方，②のループについては，A を0，1とも含んでおり，今度は A を省略し，②のループ$=B$ となる。したがって，

論理式は①と②から導びいた A と B の論理和となり，

$$f=A\vee B \tag{1.56}$$

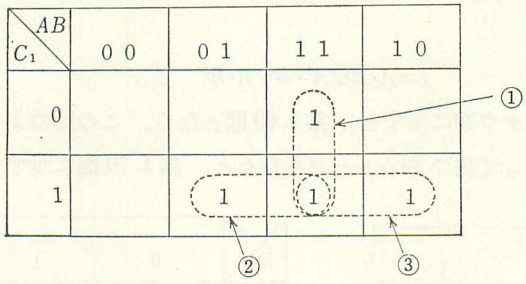
と簡素化されることになる。

〔例 1.18〕

$$f=A\cdot B\cdot C'_1\vee A'\cdot B\cdot C_1\vee A\cdot B'\cdot C_1\vee A\cdot B\cdot C_1 \tag{1.57}$$

この例は，さきに論理代数により簡素化を行なった全加算器の上位けた上げ条件を示すものである。

この論理式をカルナウ図で示し，ループを作成すると，**第1.71図**のようにな



第1.71図　$f=A\cdot B\vee B\cdot C_1\vee A\cdot C_1$

る。この図の①のループは C_1 を 0，1 とも含んでおり C_1 を省略でき，また②のループは A を 0，1 ともに含んでいるため A を省略できる。さらに③のループは B を 0,1 ともに含んでいるため B を省略でき，結果として，

$$①=A\cdot B$$

$$②=B\cdot C_1$$

$$③=A\cdot C_1$$

したがって，簡素化された論理式は，つぎのようになる。

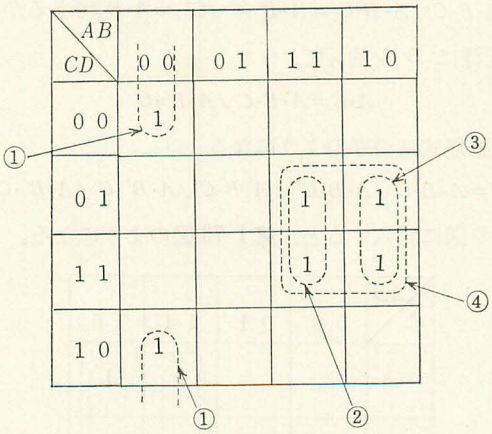
$$f=A\cdot B\vee B\cdot C_1\vee A\cdot C_1 \tag{1.58}$$

この式は論理代数で簡素化された式（1.52）と同一であり，正しい結果が得られていることがわかる。このようにカルナウ図による論理式の簡素化は機械的にできることに大きな特徴がある。

〔例 1.19〕

$$f=A'\cdot B'\cdot C'\cdot D'\vee A'\cdot B'\cdot C\cdot D'\vee A\cdot B\cdot C'\cdot D\vee A\cdot B\cdot C\cdot D\vee A\cdot B'\cdot C'\cdot D$$
$$\vee A\cdot B'\cdot C\cdot D \qquad (1.59)$$

今度は，この4変数の例を考えてみよう。この式のカルナウ図を第1.72図に示す。この図上でループ可能な1は，まず①のループである。このループはます



第1.72図 $f=A'\cdot B'\cdot D'\vee A\cdot D$

目の最上行と最下行は隣接しているものとみなすという規則にしたがって囲んだものである。また④のループは，②のループと③のループが隣接しているので，これを囲んで1つのループにしたものである。この2つのループから論理式を導びくことになるが，①のループはCを0，1の両方含んでおりCを省略し，

$$①=A'\cdot B'\cdot D'$$

となり，また④のループはBとCの2つの変数に0，1を含んでいるため，B，Cの2変数を省略し，

$$④=A\cdot D$$

となり，結果として導びかれる論理式は，つぎのようになる。

$$f=A'\cdot B'\cdot D'\vee A\cdot D \qquad (1.60)$$

〔例 1.20〕

$$f=A\cdot B\vee A\cdot C\vee A\cdot B'\cdot C' \qquad (1.61)$$

これは3変数の論理式であるが，第1項と第2項は2変数しか含んでいない。

この場合のカルナウ図への1の記入は，含まない変数を補って考えなければならない。たとえば，第1項の $A\cdot B$ は C を含んでいないため C を補い，つぎのようにして記入する。

$$A\cdot B=A\cdot B\cdot C\vee A\cdot B\cdot C'$$

(なぜならば，$A\cdot B\cdot C\vee A\cdot B\cdot C'=A\cdot B\ (C\vee C')=A\cdot B$ であるから)

また，$A\cdot C$ も同様に B を補って，

$$A\cdot C=A\cdot B\cdot C\vee A\cdot B'\cdot C$$

これで，最初の論理式はつぎのようになる。

$$f=A\cdot B\cdot C\vee A\cdot B\cdot C'\vee A\cdot B\cdot C\vee A\cdot B'\cdot C\vee A\cdot B'\cdot C' \qquad (1.62)$$

この式をカルナウ図に記入すると，**第1.73図**のようになる。

C \ AB	0 0	0 1	1 1	1 0
0			1	1
1			1	1

第1.73図　$f=A$

この図での大きな1つのループは B と C に 0，1 を含み，したがって，残る変数は A のみとなり，論理式は，次式のように簡素化されることになる。

$$f=A \qquad (1.63)$$

以上の4つの例からも明らかなように，カルナウ図は論理式の簡素化の手段として，きわめて有効である。

1.7　論理回路の構成と回路部品

前節まで and，or 等の論理回路の働きについて説明してきたが，その論理回路がどのような回路の組み合わせでできているかは触れなかった。ここでは，その論理回路の構成および記憶回路であるフリップフロップについて述べることにする。

1.7.1　回路部品

論理回路やフリップフロップの仕組みを説明する前に，それらの構成部品の主体

となっている**ダイオード**(1)と**トランジスタ**(2)について簡単にその性質を明らかにしておく。

1. ダイオード ダイオードは，以前から整流器や検波器として広く使われているもので，電流を一方向にのみ流す働き（**整流**(3)**作用**）を持っている。

これは**p形半導体**(4)と**n形半導体**(5)を**第1.74図**のように接合したもので，これを表わす図記号として，**第1.75図**に示すものを用いる。

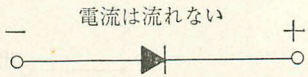
第1.74図 ダイオード

第1.75図 ダイオードの図記号

このダイオードの両端へ電圧をかけるとき，**第1.76図**のように，p形半導体側を高電位に，n形半導体側を低電位にすると，このダイオードはきわめて低い抵抗値を示し電流が流れる。逆にp形半導体側を低電位，n形半導体側を高電位にすると，ダイオードは高抵抗となり，ほとんど電流は流れない。

このように，電圧のかけ方を，p側に高電位，n側を低電位にすることを**順方向**にバイアスするといい，逆に**第1.77図**のようにp側を低電位，n側を高電位にすることを**逆方向**にバイアスするという。

第1.76図 順方向に電圧をかけた場合



第1.77図 逆方向に電圧をかけた場合

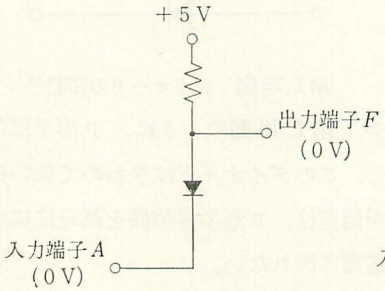
ダイオードが順方向にバイアスされたときと，逆方向にバイアスされたときとではその抵抗差が大きく，一般的には，順方向では抵抗＝0，すなわち，短絡されたのとほとんど等しく，逆方向では抵抗＝∞，すなわち，開放になったのと同じように考えてよい。

以上のことをダイオードに実際の値を与えて具体的に考えてみる。

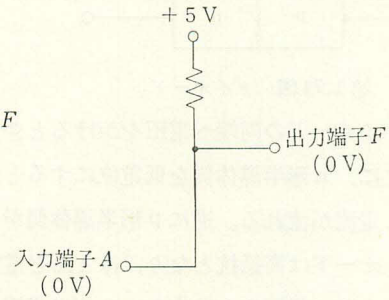
いま，**第1.78図**に示すように，ダイオードのp側に＋5Vの電圧を加え，また，入力端子Aに0Vの電圧を加えたとする。この場合，ダイオードは＋5Vと0Vで

(1) diode (2) transistor (3) rectification, commutation
(4) p-type semiconductor (5) n-type semiconductor

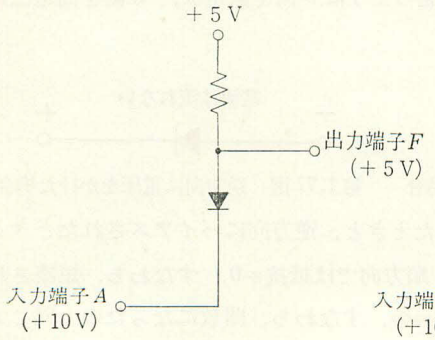
順方向にバイアスされたことになり，このダイオードは短絡され，**第1.79図**のような状態になったと考えてよい。したがって，出力端子 F には入力端子の電圧0Vがそのまま現われることになる。つぎに，**第1.80図**のように入力端子 A に+10Vが加えられた場合を考えてみる。この場合は，ダイオードは逆方向にバイアスされ，ダイオードは開放されたと同じことで，**第1.81図**のように考えることができる。この状態では，出力端子 F にはp側の印加電圧+5Vそのままに現われる。以上の



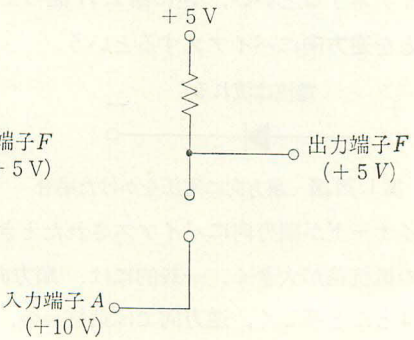
第1.78図



第1.79図



第1.80図



第1.81図

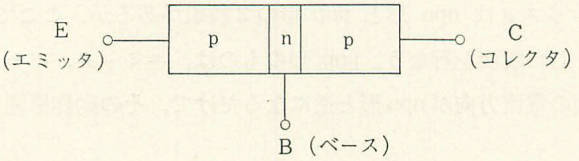
第1.17表　ダイオードと電圧の関係

p側電圧〔V〕	n側(入力)電圧〔V〕	バイアス方向	出力電圧〔V〕
+5	0	順方向	0
+5	+10	逆方向	+5

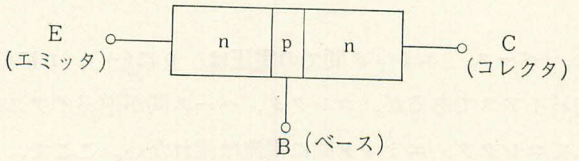
ことを表にすると，**第1.17表**のようになる。

2.　トランジスタ　　トランジスタは，日常にラジオやテレビジョン等の増幅回路に広く使われている回路素子である。

トランジスタは，p形半導体とn形半導体を3つ接合して作成したものであり，その接合の方法には，**第1.82図**に示すように，pnp の順序に，n形半導体をサンドイッチにする方法と，**第1.83図**の npn のように，p形半導体をサンドイッチに



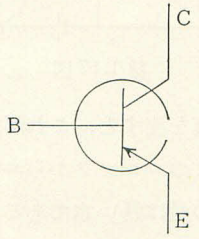
第1.82図　pnp 形トランジスタ



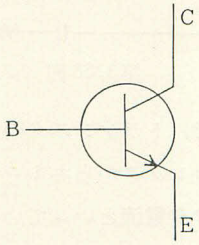
第1.83図　npn 形トランジスタ

する方法がある。この2つのトランジスタのうち，第1.82図のn形半導体をサンドイッチにするものを，**pnp 形トランジスタ**といい，図記号として**第1.84図**のものを使用している。もう一方のp形半導体をサンドイッチにするものを，**npn 形トランジスタ**といい，その図記号は**第1.85図**に示すとおりである。

この図の3つの端子に記されている B，C，E は，第1.82図，第1.83図の B，



第1.84図　pnp 形トランジスタの図記号



第1.85図　npn 形トランジスタの図記号

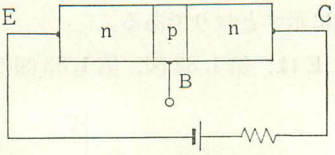
C, E に対応しており，それぞれ，**ベース**(1)，**コレクタ**(2)，**エミッタ**(3)と呼ばれる。また各部に流れる電流をベース電流（I_B），コレクタ電流（I_C），エミッタ電流（I_E）という。エミッタについている矢印は，エミッタ，ベース間，エミッタ，コレクタ間の**電流の方向**を示している。

このトランジスタの動作はダイオードのように簡単でなく，かなり複雑なものであるが，ここではごく大ざっぱにその働きを述べることにする。

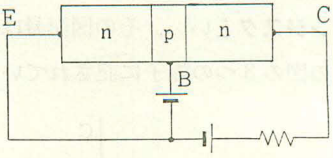
また，トランジスタは npn 形と pnp 形の2種類があるが，ここでは，npn 形のものを主体として説明を行なう。pnp 形のものは，エミッタ，ベース間，エミッタ，コレクタ間の電流方向がnpn形と逆になるだけで，その動作原理はまったく同じである。

さて，この npn形 トランジスタに**第1.86図**のように電圧を加えた場合を考えてみよう。

この状態では，ベース，エミッタ間での電圧は，pに(＋)，nに(－) がかかったことで順方向バイアスであるが，コレクタ，ベース間が逆バイアスになっているので，結果としてコレクタ，エミッタ間に電流は流れない。ここで，もう1つの電圧をベース，エミッタ間に**第1.87図**のように加えてみる。今度は，ベース，エミッタ間の電圧は順方向であるので，ベースからエミッタに電流が流れる。この電流を**ベース電流**といっている。



第1.86図



第1.87図

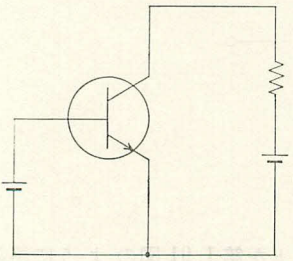
ところで，トランジスタの性質として，ベース電流を流すと，これが誘い水の役目をしてコレクタからエミッタへの電流が流れる。このコレクタからエミッタへ電流を**コレクタ電流**といっている。このコレクタ電流の量は誘い水であるベース電流

(1) base　(2) collector　(3) emitter

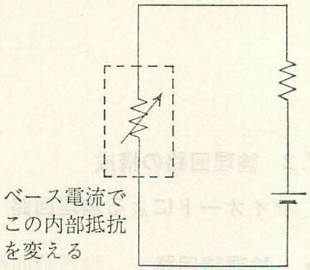
の量によって左右される。この誘い水（ベース電流）が多ければコレクタ電流も多く，ベース電流が少なければコレクタ電流も少なくなる。

しかも，このベース電流とコレクタ電流の比は，5：95以上になっていて，小さいベース電流で大きなコレクタ電流を制御しうることになり，ここにいわゆる**増幅作用**を生ずることになる。第1.87図をトランジスタの図記号を使って書き替えると第1.88図のようになる。

ベース電流の増減によりコレクタ電流が増減することは，ベース電流はコレクタ，エミッタ間の**内部抵抗**(1)を変える機能を有することである（第1.89図参照）。すなわち，ベース電流が流れていないときのトランジスタの内部抵抗は無限大に近いほど大きく，ベース電流を徐々にふやしていくと内部抵抗も徐々に小さくなり，ベース電流がある大きさに達すると内部抵抗はほとんど0になる。この内部抵抗がほとんど0になり，コレクタ電流が一番多く流れる状態をトランジスタが**飽和領域**にあるという。逆に，内部抵抗が無限大に近く，コレクタ電流が流れない状態を**しゃ断領域**にあるという。この，しゃ断領域と飽和領域の間で，コレクタ電流が除々に増減する部分を**活性領域**と呼び，普通のラジオやテレビジョンで使用するトランジスタは，この活性領域で動作している。しかし，計算機は，1か0かを扱うものであり，その中間は必要ないので電流が流れるか流れないか，すなわち，飽和領域としゃ断領域を1，0に対応させて使っている。このような使い方をするトランジスタを**スイッチング・トランジスタ**という。



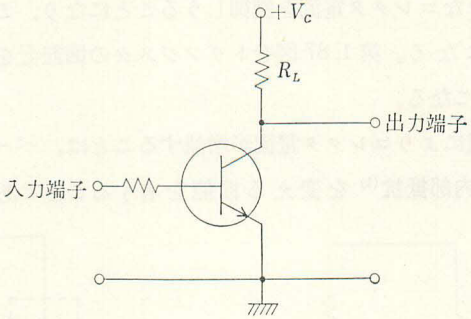
第1.88図



第1.89図

(1)　internal resistance

トランジスタの入出力の使い方にもいろいろあるが，一般に，**第1.90図**のように，入力をベースから入れ，コレクタから出力を取る使い方が多い。このようなトランジスタの入出力の使い方を**エミッタ接地形**(1)という。



第1.90図

1.7.2　論理回路の構成

1.　ダイオードによる論理回路

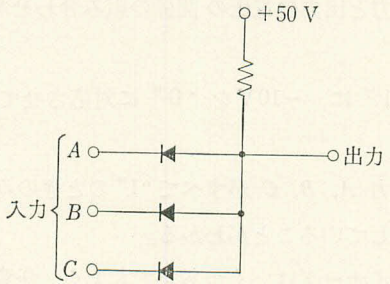
a.　論理積回路　論理積回路はダイオードを**第1.91図**のように組み合わせて作ることができる。いま，この回路の入力A，B，C に＋5V，−10V が**第1.92図**のように加わった状態を考えてみる。

まず，図(**a**)では3入力とも＋5V であり，ダイオードのp側が＋50V であるため，すべてのダイオードが順方向にバイアスされ，出力には入力電圧＋5Vが現われることになる。

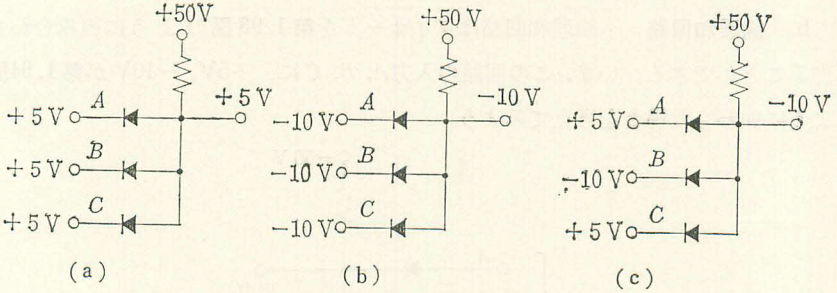
つぎに，図(**b**)では3入力が−10Vで，これも順方向に変わりなく出力には入力電圧−10Vが現われる。

問題は最後の図（**c**）である。これはB入力に−10V，他に＋5V がかかっている。ここで，図(a)と同じようにAのダイオードが順方向だとすると，出力は＋5Vになる。しかし，Bのダイオードのn側は−10V であるため，このダイオードはp側に＋5V，n側に−10V，すなわち，順方向にバイアスされており，したがって，出力にはB入力の−10Vが現われる。出力が−10Vになると，A，C のダイ

(1)　emitter earth



第 1.91 図 ダイオードによる論理積回路



第 1.92 図

オードは逆方向にバイアスされたことになり，結果として出力は−10Vとなる。

このことは−10Vがどの入力にあっても同じことであり，1つでも−10Vの入力があると出力は，その低い方の電圧に引きずられて−10Vとなる。

第 1.18 表 論理積回路の電圧

入力電圧〔V〕			出力
A	B	C	電圧
−10	−10	−10	−10
−10	−10	+ 5	−10
−10	+ 5	−10	−10
−10	+ 5	+ 5	−10
+ 5	−10	−10	−10
+ 5	−10	+ 5	−10
+ 5	+ 5	−10	−10
+ 5	+ 5	+ 5	+ 5

第 1.19 表 論理積回路の真理値表

入力			出力
A	B	C	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

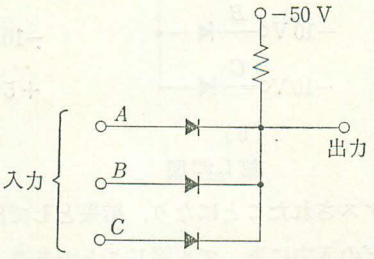
この回路における入力と出力の電圧の関係の組み合わせを示すと，**第1.18表**のようになる。

この表の＋5Vを“1”に，－10Vを“0”に対応させて表を書き替えると，**第1.19表**のようになる。

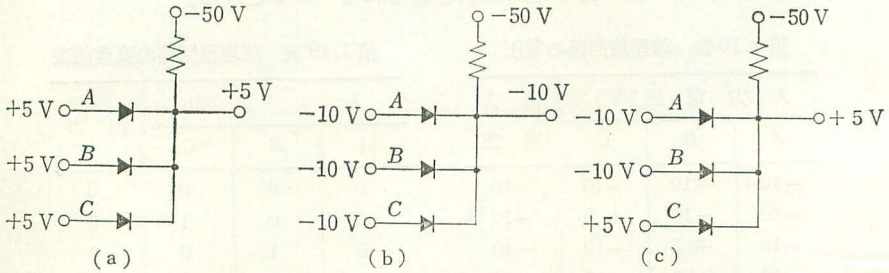
この表でみると，入力 *A*, *B*, *C* がすべて“1”のときのみ出力が“1”となり，論理積回路の条件を満足していることがわかる。

この場合，入力のダイオードはいくつあってもよく，任意の入力数の論理積回路をダイオードを使って作成することができる。

b.　論理和回路　論理和回路はダイオードを**第1.93図**のように組み合わせて作ることができる。いま，この回路の入力*A*, *B*, *C*に，＋5V，－10Vが**第1.94図**のように加わった場合を考えてみよう。



第1.93図　ダイオードによる論理和回路



第1.94図

図(**a**)では，3入力とも＋5Vであり，すべてのダイオードは順方向にバイアスされ，出力には入力の＋5Vが現われることになる。

つぎの図(**b**)は，3入力が－10Vに変わっただけで，図(a)と同じく出力には

入力の−10Vが現われる。

最後の図(c)は，1つだけ+5V，他は−10Vで，この状態では3つのダイオードとも順方向に見えるがそうではない。もし，ダイオードAが順方向だとすると，出力にはAの入力−10Vが現われることになる。しかし，出力が−10Vになると，ダイオードCの入力は+5Vであるため，+5Vと−10Vでは+5Vの方が高く，ダイオードCが順方向になり，出力はCの入力+5Vになる。

出力が+5Vになれば，ダイオードA，Bは逆方向となり，結果として+5Vが出力に現われることになる。こうして，入力のどれかが+5Vであれば出力は，その高い電圧に引きずられて+5Vになる。

この回路の入力と出力の電圧関係を**第1.20表**に示す。これを論理積回路のときと同じように+5Vを“1”，−10Vを“0”に対応させて書き替えると，**第1.21表**のようになり，入力A，B，Cのいずれか1つが“1”のとき出力は“1”となり論理和条件を満足している。この場合も，論理積回路と同様に入力はいくつあってもよい。

第1.20表 論理和回路の電圧

入力電圧〔V〕			出力
A	B	C	電圧
−10	−10	−10	−10
−10	−10	+ 5	+ 5
−10	+ 5	−10	+ 5
−10	+ 5	+ 5	+ 5
+ 5	−10	−10	+ 5
+ 5	−10	+ 5	+ 5
+ 5	+ 5	−10	+ 5
+ 5	+ 5	+ 5	+ 5

第1.21表 論理和回路の真理値表

入力			出力
A	B	C	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

c. 正論理と負論理 前述の論理積回路と論理和回路は，すべて+5Vを“1”−10Vを“0”に対応させて考えてきた。しかし，この対応のさせ方はあくまでとり決めであるので，逆に−10Vを“1”，+5Vを“0”と考えてもなしつかえない。

この対応のさせ方で，電圧の高い方を“1”，低い方を“0”にするのを正論理といい，逆に低い電圧を“1”，高い電圧を“0”に対応させるのを負論理という。さきに，ダイオードで作成した回路は正論理の論理積回路および正論理の論理和回路

である。

ではここで，さきの論理積回路，論理和回路を負論理にして考えてみよう。

論理積回路の入出力電圧の関係を示す第1.18表を，−10Vを“1”，+5Vを“0”に対応させて書き替えてみると**第1.22表**のようになる。

第1.22表　論理和回路の真理値表

入力			出力
A	B	C	
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

第1.23表　論理積回路の真理値表

入力			出力
A	B	C	
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

この第1.22表を見ると，これは論理和回路の表を示す第1.21表の順序を逆にしただけでまったく同じものである。このことから，正論理と負論理の関係は，つぎのようにいえる。

正論理の論理積回路＝負論理の論理和回路　　(1.64)

また，論理和回路の入出力電圧の関係を示す第1.20表を負論理として書き替えると**第1.23表**のようになる。これも正論理の論理積回路を示す第1.19表とまったく同じであり，正論理と負論理と関係はつぎのようになる。

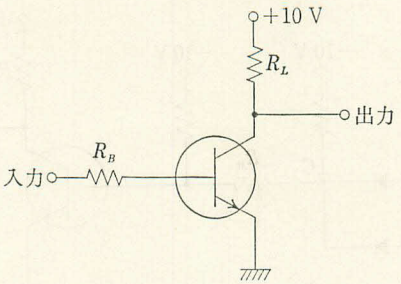
正論理の論理和回路＝負論理の論理積回路　　(1.65)

以上のように，正論理と負論理ではandとorが逆になり，厳密には，論理積回路，論理和回路はどちらの論理のものであるかを区別していう必要がある。しかし，以下ここではとくにことわらない限り，正論理の回路を考えることにする。

2.　トランジスタによる論理回路

a.　否定回路　信号を反転する否定回路は，トランジスタを**第1.95図**のように使用することにより簡単に作成できる。

いま，入力のベースに0Vが加えられたとすると，ベースが0Vではベース電流



第1.95図 トランジスタによる否定回路

は流れず，したがって，このトランジスタはしゃ断領域にあり，コレクタ電流は流れない。そのときのコレクタからの出力電圧はコレクタの電源電圧＋10Vがそのまま現われる。

つぎに，入力に＋10Vを加えた場合を考えると，ベースに接続されているベース抵抗 R_B を適当に選んでおけば，入力に加えられた＋10Vにより，このトランジスタがちょうど飽和領域で動作するようにベース電流を流すことができる。トランジスタが飽和領域にあるということは，内部抵抗がほとんど0となり，コレクタ，エミッタ間の電圧降下はきわめて小さい。すなわち，コレクタの出力電圧はほとんど0Vとなる。

以上の入力，出力の電圧関係を示すと**第1.24表**のようになる。

この表の＋10Vを“1”，0Vを“0”に対応させて書き替えたのが，**第1.25表**であり，このトランジスタ回路が否定回路の働きをしていることは明らかであろう。

第1.24表 否定回路の電圧

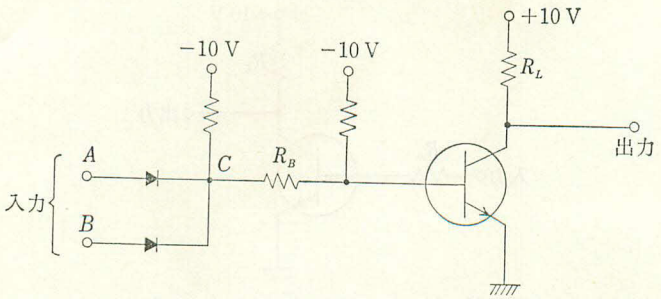
入力電圧〔V〕	出力電圧〔V〕
0	＋10
＋10	0

第1.25表 否定回路の真理値表

入　力	出　力
0	1
1	0

b. nor 回路(1)　いままでに，ダイオードによる論理和回路とトランジスタによる否定回路を説明してきたが，この論理和と否定回路を組み合わせて使うことが多い。この論理和と否定回路を組み合わせた回路を，not と or という意味で **nor（ノア）回路**という。この nor 回路は，**第1.96図**のようになる。

(1) “nor” circuit



第1.96図　ダイオードとトランジスタによる nor 回路

いま，この入力 A，B には，0 Vまたは+10Vが加えられるものとし，論理和回路の出力 C 点が+10Vのとき，否定回路のトランジスタが飽和状態になるよう回路を組んでおけば，この回路の入力電圧の関係は，**第1.26表**のようになる。

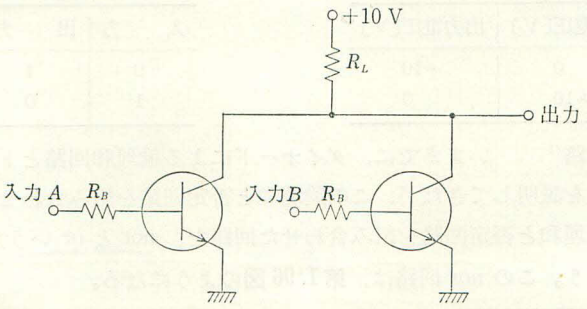
第1.26表　nor 回路の電圧

入力電圧〔V〕		出力電圧〔V〕
A	B	
0	0	+10
0	+10	0
+10	0	0
+10	+10	0

第1.27表　nor 回路の真理値表

入力		出力
A	B	
0	0	1
0	1	0
1	0	0
1	1	0

この表を前のように，0 Vを“0”，+10Vを“1”として書き替えたものが**第1.27表**である。この第1.27表の出力を見ると，論理和回路であれば“1”となる



第1.97図　トランジスタによる nor 回路

べきところが“0”に，逆に“0”になるべきところが“1”になっており，nor 回路であることを示している。

nor 回路はトランジスタのみで作成することができる。そのためには**第1.97図**に示すように，同じ規格のトランジスタを並列に接続すればよい。この nor 回路は，さきの否定回路を2つ並べたものと考えてよく，トランジスタが2個ともしゃ断状態であれば出力は電源電圧＋10Vであり，1つでも飽和状態になれば出力は0Vとなる。この回路の入力，出力の電圧関係を**第1.28表**に示す。この表を＋10Vを“1”，0Vを“0”に対応させて書き替えたものが**第1.29表**である。

第1.28表 nor 回路の電圧

入力電圧〔V〕		出力電圧〔V〕
A	*B*	
0	0	+10
0	+10	0
+10	0	0
+10	+10	0

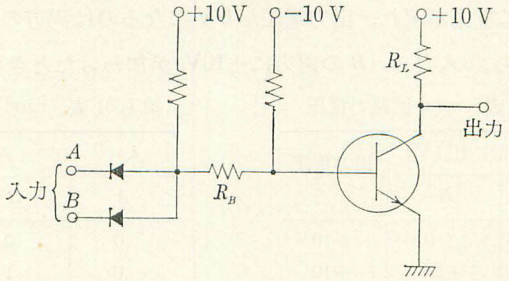
第1.29表 nor 回路の真理値表

入力		出力
A	*B*	
0	0	1
0	1	0
1	0	0
1	1	0

この表は，さきのダイオードとトランジスタで作成した nor 回路の第1.27表とまつたく同じものである。

c. nand 回路(1)　nand(ナンド) 回路は，not and 回路を意味し，論理積回路の出力を反転したものである。

nand 回路も nor 回路と同様に，ダイオードによる and 回路に not 回路を接続



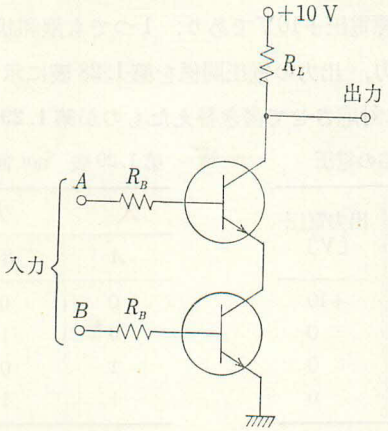
第1.98図　ダイオードとトランジスタによる nand 回路

(1)　“nand” circuit

しても作成できるし，また，トランジスタのみで作成することもできる。

まず，ダイオードとトランジスタを組み合わせた nand 回路の例を第1.98図に示す。この図は，いままで述べた論理積回路と否定回路が接続されているものであり，その動作は容易に理解できるものと思われるので説明を省略する。

つぎに，トランジスタのみの nand 回路の例を第1.99図に示す。



第1.99図　トランジスタによる nand 回路

nor 回路はトランジスタを並列に接続したものであったのが，この nand 回路は直列に接続してある。いま，入力 A，B には＋10V か 0 V の電圧が加わり，入力に＋10V の電圧が加わったときには，そのトランジスタは飽和状態になり，また，入力に 0 V が加わったときは，しゃ断状態になるように各回路定数を定めておけば，出力抵抗 R_L に電流が流れ，出力電圧が 0 V になるのは両方のトランジスタが飽和状態，すなわち，入力 A，B の両方に＋10V が加わったときのみとなる。この入

第1.30表　nand 回路の電圧

入力電圧〔V〕		出力電圧〔V〕
A	B	
0	0	+10
0	+10	+10
+10	0	+10
+10	+10	0

第1.31表　nand 回路の真理値表

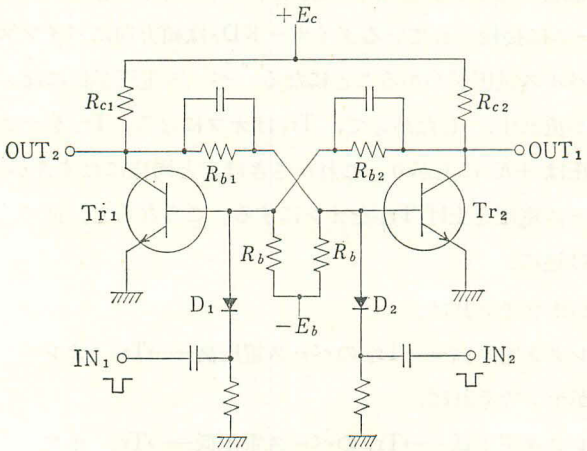
入力		出力
A	B	
0	0	1
0	1	1
1	0	1
1	1	0

力，出力の電圧関係を示すものが**第 1.30 表**である。

これをまた，＋10Vを“1”，0Vを“0”に対応させて書き換えると**第 1.31 表**のようになる。この表の出力をみると and 回路であれば，“1”となるべきところが“0”に，逆に“0”になるべきところが“1”になり，nand 回路であることを示している。

3. フリップフロップ回路　レジスタの構成回路として広く使われているフリップフロップ回路は，**第 1.100 図**に示すように，まったく同規格のトランジスタ Tr_1，Tr_2 を2個対称的に接続したものであり，入力は各トランジスタのベースに加え，出力はコレクタから取り出している。

そして，この2個のトランジスタは両方とも，**飽和状態**(オン)(1)，または両方と



第 1.100 図　フリップフロップ回路

もしゃ断状態（オフ）(2) になることのないように回路定数が定められている。

いま，Tr_1 がオン，Tr_2 がオフだとすれば，オンの Tr_1 の内部抵抗は0に等しく，したがって，Tr_1 のコレクタ電圧はほとんどアース電位（0V）である。この Tr_1 のコレクタ電圧（0V）は，たすきがけになっている抵抗 R_{b1} を経て Tr_2 のベースにかかり，Tr_2 のベース電圧を低電位とし，Tr_2 をオフにする役目を果たしている。一方，オフである Tr_2 を見ると，この Tr_2 に電流が流れていないため，Tr_2 の

(1) on　(2) off

コレクタ電圧は電源電圧 E_c に等しい。このことは，この電圧がやはりたすきがけ抵抗 R_{b2} を経て Tr_1 のベースにかかり，Tr_1 のベース電圧を高くして Tr_1 をオンにする働きをもっている。以上のことを整理してみると，

① Tr_1 がオンであれば，

Tr_1 のコレクタ電圧低⟶Tr_2 のベース電圧低⟶Tr_2　オフ

② Tr_2 がオフであれば，

Tr_2 のコレクタ電圧高⟶Tr_1 のベース電圧高⟶Tr_1　オン

となり，Tr_1，Tr_2 がお互いに影響し合って Tr_1 がオン，Tr_2 がオフの状態で安定している。

いま，この状態のときに，Tr_1 の入力端子 IN_1 に負の電圧パルスを与えてやると，Tr_1 のベースに接続されているダイオード D_1 は順方向にバイアスされ，Tr_1 のベースは負のパルス電圧がかかることになる。ベース電圧が負になったのでは Tr_1 にベース電流が流れず，したがって，Tr_1 はオフになる。Tr_1 がオフになると Tr_1 のコレクタ電圧は $+E_c$ に上がり，これがさきほどと同様にたすきがけ抵抗 R_{b1} を経て Tr_2 のベース電圧を上げ Tr_2 をオンにする。こうなると，前の Tr_1 オン，Tr_2 オフのときとは逆に，

① Tr_1 がオフであれば，

Tr_1 のコレクタ電圧高⟶Tr_2 のベース電圧高⟶Tr_2　オン

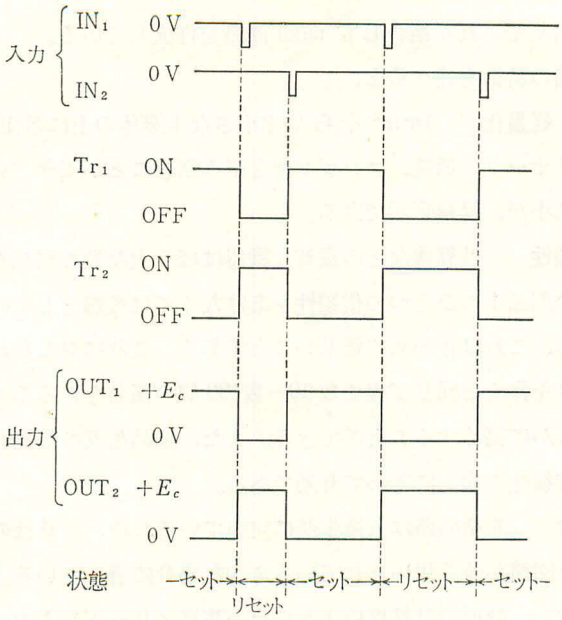
② Tr_2 がオンであれば，

Tr_2 のコレクタ電圧低⟶Tr_1 のベース電圧低⟶Tr_1　オフ

となり，Tr_1 がオフ，Tr_2 がオンの状態に安定するのである。

すなわち，Tr_1 がオン，Tr_2 がオフのとき Tr_1 の入力に負の電圧パルスを与えれば，Tr_1，Tr_2 の状態が反転し，Tr_1 がオフ，Tr_2 がオンになる。同様にして，Tr_1 がオフ，Tr_2 がオンのときに Tr_2 の入力に負の電圧パルスを与えれば，Tr_1 がオン，Tr_2 がオフに反転する。

いま，Tr_1 がオンのときを**セット状態**，Tr_2 がオンのときを**リセット状態**とすれば，セット，リセット時における Tr_1，Tr_2 の動作状態および出力電圧の関係は**第1.101図**のとおりである。



第1.101図 フリップフロップの状態

この図からもわかるように，Tr_2 への入力信号が**セット信号**であり，Tr_1 への入力信号が**リセット信号**である。

また，セット出力は OUT_1，リセット出力は OUT_2 となる。

なお，セット，リセットの両入力に同時に信号がきた場合は，このフリップフロップの状態がどうなるかは**不定**である。

1.7.3 集積回路(1)

従来使われている回路は，各部品を別々に作り，それぞれの部品を電気的に接続して，ある機能を持った回路にするという方法が取られていた。

しかし最近では，これらの部品を1まとめにして1つの半導体上に作成する**集積回路**が登場してきた。たとえば，nand 回路を作るには，いままではトランジスタ，ダイオード，抵抗を別々に作り，あとでこれを適当に接続していたのであるが，集積回路ではある半導体の上にトランジスタやダイオードとともに抵抗もいっ

(1) integrated circuit：略して IC

しよに作り，内部でこれを結合して nand 回路を作成している。

この集積回路の特徴を述べると，

（1） **小形，軽量化**　1mm² ぐらいの小さな半導体の上に数10個以上のトランジスタやダイオード，抵抗，コンデンサを組み込むことができ，いままでの回路に比べ飛躍的に小形，軽量化ができる。

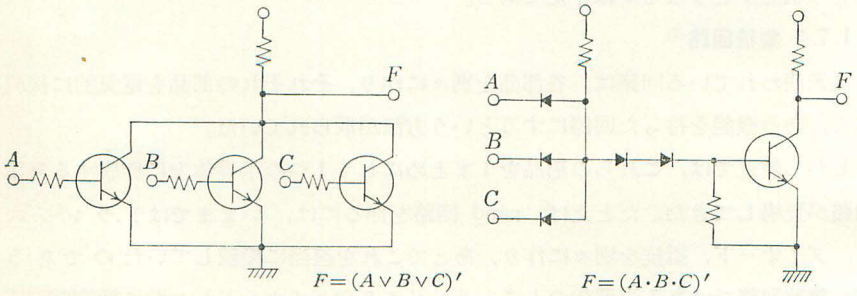
（2） **高信頼性**　計算機などの複雑な機器はぼう大な数の部品を必要としており，その個々の部品1つひとつの信頼性をあげなくては機器としての信頼性は上がらない。しかし，これはきわめて難しいことである。これに対し集積回路では1つのトランジスタを作ると同じ工程で数10～数100個の部品を作るので，全体としての信頼性をはるかに高くすることができる。また，部品相互の接続個所も少なくなり，この点も信頼性の向上にきわて有効である。

（3） **経済性**　集積回路は大量生産に向いているため，計算機の論理回路のように同じような回路が多く用いられているような場合に適している。

（4） **高速化**　最近の計算機のようにその演算スピードが1ナノ秒 (10^{-9}s) 単位で争われるようになると，わずかな配線長による伝搬遅れが問題になってくる。この点からも集積回路による小形化の特徴が生かされる。

こうして，計算機には，集積回路が広く採用されており，近いうちにほとんどの回路が集積化されるものと考えられる。

現在作成されている集積回路の例を**第1.102図**に示す。



（a） RTL 回路　　(b) DTL 回路

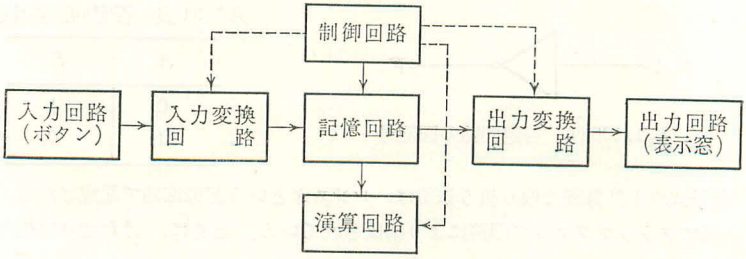
第1.102図

1. RTL 回路[1]　抵抗とトランジスタにより作られた回路で，機能は，$F=(A\vee B\vee C)'$ すなわち，nor 回路である。

2. DTL 回路[2]　ダイオードとトランジスタにより作られた回路で，機能は $F=(A\cdot B\cdot C)'$ すなわち，nand 回路である。

1.8 摘　　要

1. 電子式卓上計算機は，つぎの7つの部分から構成されている。
（1） 入力回路，（2） 入力変換回路，（3） 記憶回路，（4） 演算回路，
（5） 制御回路 （6） 出力変換回路，（7） 出力回路，

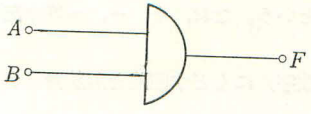


第1.103図

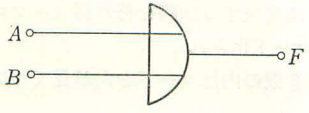
2. 計算機での数値の表現には2進法が用いられている。また，負数は補数により表現されている。

3. 2進法の演算は10進法の演算と同じ考え方で行なわれる。ただし計算機では，減算を補数加算の方法で行なう点が異なる。

4. 演算回路の基本となるものに，論理積回路，論理和回路，否定回路がある。
（1） 論理積回路　論理積回路は入力がすべて1のときのみ出力に1が出る回路である。



第1.104図　論理積回路の図記号



第1.105図　論理和回路の図記号

（1） resistor transistor logic circuit：略して RTL circuit
（2） diode transistor logic circuit：略して DTL circuit

（2）論理和回路　論理和回路は，入力のいずれか1つが1であれば，出力に1がでる回路である。

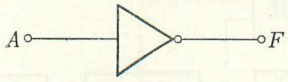
第1.32表　論理積回路真理値表

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

第1.33表　論理和回路真理値表

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

（3）否定回路　否定回路は，入力が1のときは出力を0，入力が0のときは出力を1というように，入力信号を反転して出力する回路である。



第1.106図　否定回路の図記号

第1.34表　否定回路真理値表

A	F
0	1
1	0

5.　電子式卓上計算機で取り扱う数値は，レジスタという記憶回路で記憶される。レジスタは，一般にフリップフロップ回路により構成されている。とくに，けた送り機能を持たせたレジスタを送りレジスタという。

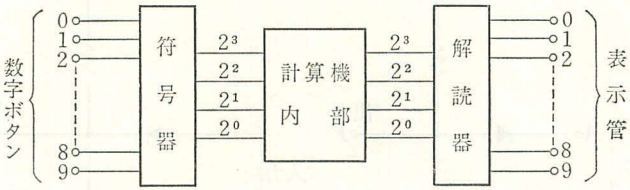
6.　演算回路は加算器を中心として構成されており，すべての演算は加算（補数加算を含めて）を基本にして行なわれる。そしてその制御は，制御回路によりなされている。

7.　加算の方法には，下位から1けたずつ順に加算していく直列加算方式と，何けたかを同時に加算する並列加算方式とがある。一般に，電子式卓上計算機では直列加算方式が用いられている。

8.　計算機には，一定周期で発生するタイミング・パルスの制御のもとに動作する同期式と，ある動作の終了がすぐにつぎの動作を誘起する非同期式の計算機がある。現在，ほとんどの計算機は同期式である。

9.　数字ボタンを押すことにより入力される0～9の数を計算機で取り扱える内部コード（2進数）に変換する回路を符号器（エンコーダ）という。なお，＋，－，＝等の記号もこの符号器でコード化される。

10.　計算機の内部コードを人が見てわかる形（数字）にもどす回路を解読器（デコーダ）という。



第1.107図　符号器と解読器

11. 論理式の変形，簡素化等には論理代数（ブール代数）が用いられる。この論理代数の基本的な諸公式は，つぎ次のとおりである。

（論理和）

$A \vee 0 = A.\quad A \vee 1 = 1,\quad A \vee A = A$

（論理積）

$A \cdot 0 = 0,\quad A \cdot 1 = A,\quad A \cdot A = A$

（交換の定理）

$A \vee B = B \vee A,\quad A \cdot B = B \cdot A$

（結合の定理）

$A \vee (B \vee C) = (A \vee B) \vee C,\quad A \cdot (B \cdot C) = (A \cdot B) \cdot C$

（分配の定理）

$A'' = A,\quad A \vee A' = 1,\quad A \cdot A' = 0$

（ド・モルガンの定理）

$(A \vee B)' = A' \cdot B',\quad (A \cdot B)' = A' \vee B'$

12. 論理式を図形上で簡素化する方法の1つとしてカルナウ図がある。

13. 論理回路において，論理の“1”，“0”と電位の高低の対応のさせ方には，つぎの2つがある。

（1）　正論理　　高電位→1，　低電位→0

（2）　負論理　　低電位→1，　高電位→0

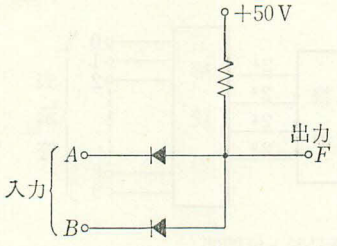
14. 正論理と負論理の間には，つぎの関係がある。

正論理の論理積回路＝負論理の論理和回路

正論理の論理和回路＝負論理の論理積回路

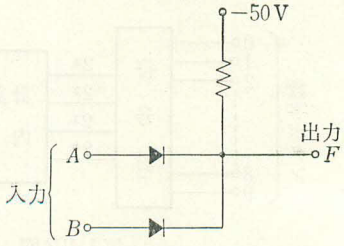
15. 論理積回路，論理和回路はダイオードやトランジスタを使って作成される。

そのうち，ダイオードを用いた論理積回路，論理和回路をつぎに示す。



$F=A\cdot B$

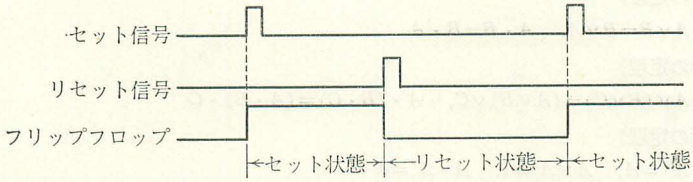
第1.108図　論理積回路



$F=A\vee B$

第1.109図　論理和回路

16. フリップフロップ回路は，2つの安定状態（セットおよびリセット）を持ち，この2つの安定状態をそれぞれ“1”および“0”に対応させることにより，1ビットを記憶することができる。このフリップフロップ回路の2つの状態と入力信号との関係は，つぎのとおりである。



第1.110図

17. 最近の計算機では集積回路（IC）が多く使われるようになった。この集積回路は，従来のように種々の回路素子を個々に集めてある回路を作るのではなく，1つの半導体上にダイオードやトランジスタ，抵抗，コンデンサ等を作り，これを内部で結合したものである。

これにより，いままでの回路に比べて小形，軽量，高信頼性等の利点があり，今後の計算機のほとんどが集積回路化されることが予想される。

1.9 練 習 問 題

1. つぎの10進数を2進数で表わせ。

（1） 14,　（2） 259,　（3） 362

2. つぎの2進数を10進数で表わせ。

（1） 10001,　（2） 111001,　（3） 101010

3. つぎの10進数を2進化10進数で表わせ。

（1） 14,　（2） 259,　（3） 362

4. つぎの2進加算を行なえ。

（1） 011＋101, （2） 1110＋0101, （3） 0110110＋0001010

5. つぎの2進減算を補数加算の方法で行なえ。

（1） 1001−0011, （2） 1100−1010

6. つぎの論理式を満足する論理回路を示せ。

（1） $F=A'\cdot B\vee B'\cdot C\vee C'\cdot A$

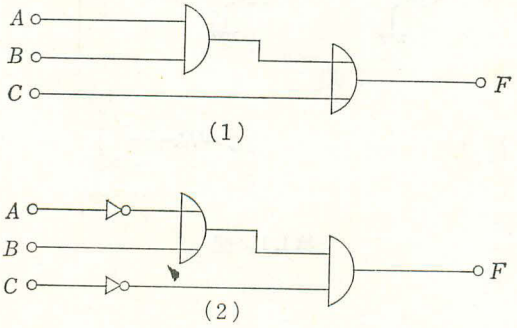
（2） $F=A'\cdot B'\cdot C'\vee A'\cdot B'\cdot C\vee A\cdot B\cdot C\vee A\cdot B'\cdot C$

7. つぎの真理値表から出力 F_1, F_2 を得るための論理式と論理回路を示せ。

第1.35図 真 理 値 表

入力		出力	
A	B	F_1	F_2
0	0	1	1
0	1	0	1
1	0	0	1
1	1	1	0

8. つぎの論理回路の論理式と真理値表を作成せよ。



第1.111図

9. つぎの論理式をカルナウ図を使って簡素化し，その簡素化された式を満足する論理回路を示せ。

（1） $A\cdot B\vee A\cdot B\cdot C\vee A\cdot B'\cdot C$

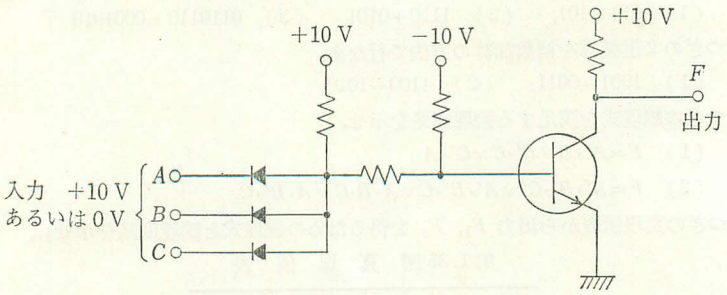
（2） $A\cdot B\cdot C\cdot D\vee A'B\cdot C\cdot D\vee A\cdot B'\cdot C\cdot D\vee A\cdot B\cdot C\cdot D'$

10. ダイオードを用いて，つぎに示す論理式の論理回路を作成せよ。

（1） $F=A\cdot B\vee C$, （2） $F=(A\vee B\vee C)\cdot D$

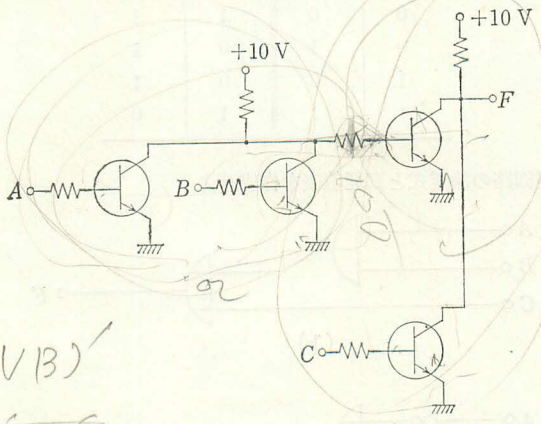
11. つぎの図の回路の入力，出力の電圧関係と負論理での真理値表を示せ。また，この回

路は負論理では，nand，nor どちらの回路であるか。



第1.112図

12.　つぎの回路の出力を表わす論理式を示せ。



第1.113図

第2章　小形電子計算機

2.1 概　　説

第1章では，電子式卓上計算機を対象として，機械が計算するのはどのような原理に基づいて行なわれているかを説明した。

本章では，いよいよ磁気テープや印刷機を備えた「本格的」な計算機の説明にはいる。ここで，「本格的」な電子計算機というものが，第1章の電子式卓上計算機とどのように異なるかについて，概念的に述べてみよう。

電子式卓上計算機では，計算すべき数値と計算方法を交互に与えてやらなければならない。すなわち，まず数値のボタンを押し，そのあとで [+] のボタンを押して「加えよ」と命令し，また，数値のボタンを押してから [×] のボタンを押して「かけろ」と命令しなければならない。これはロボットにたとえると，まず，「肉屋へ行け」と命令し，肉屋へ行ったところで「牛肉500g買え」といい，つぎに「魚屋へ行け」と命令し，魚屋へ行ったところで「ひらめを5匹買え」というように，1つずつ命令を与えてやらないと働らいてくれない，かなり低級ロボットである。

これに対して，本格的な電子計算機では，あらかじめ計算する順序を覚えさせておけば，あとは，数値だけ与えてやると，計算機は教えられた手順どおりに計算をやってくれる。これを先のロボットの例で考えると，この場合のロボットは，最初肉屋へ行って買物をし，つぎに魚屋へ行って買物をするという順序を最初に命令しておけば，取り消しをしないかぎりはその命令を覚えていて，あとは好きなときに牛肉 500g，ひらめ5匹という「買うもの」だけをいってやると，命令されたとおりの順番で店屋をまわり買物をしてくれるようなもので，前のロボットに比べると，かなり高級なロボットであるといえる。

なお，本章は「小形電子計算機」という題名を用いているが，第3章の高級な電子計算機とは基本的な構成や機能において大きく変わるところはない。しいて区別をつけるとすれば，後者がデータの高速処理を行なったり，通信回線を接続して即

時に，多種類の処理を行なうために必要な機能を持っているとか，人間が電子計算機と容易に情報の授受を行なうことができるように，新らしい装置が接続できるなどの点である。

2.1.1　電子計算機の用語

1.　プログラム(1)　　電子計算機で計算を行なう場合には，

（1）いつ数値を入力するか。

（2）入力された数値をどこに記憶するか。

（3）いつ，どのような計算をするか。

（4）中間結果をどこに記憶するか。

（5）いつ計算を止めるか。

（6）最終結果をいつ，どのような形で外へ出すか。

といった手順を明確にしておかねばならない。こうした手順は，あらかじめ人間が作っておき，計算を始める前に「計算機に対する命令」として電子計算機に覚え込ませておく必要がある(2)。

このような一連の命令群を**プログラム**といい，プログラムを作ることを**プログラミング**(3)という。また，このようにプログラムを記憶しておく方式の電子計算機を，**プログラム記憶式計算機**(4)という。

卓上計算機は，もっぱら数値の計算をするために使用されるものであるが，本章で述べるような電子計算機は，計算以外にも多くの目的に使われる。たとえば，名前をアルファベット順に並べるような場合にも用いられる。この場合，電子計算機に与えるのは数値ではなくて，アルファベットの名前である。これを**数値**(5)より広い意味で**データ**(6)という言葉で表わしている。また，名前をアルファベット順に並べるという仕事を**計算する**といったのでは不適確になるので，今後はより一般的に「**処理する**」という言葉を用いることにする。もちろんデータには数値も含まれており，「処理する」という言葉には「計算する」ことも含まれている。

（1）program
（2）この方式は，フォン・ノイマン（von Neuman）が提唱したものである
（3）programming　（4）stored program computer　（5）numerical value
（6）data

2. 情報[1]とデータ　1で数値より広い意味でデータという言葉を用いたが，これとよく似た言葉で**情報**という言葉もよく使われるので，ここで，その区別をしておく。

いま，A高校の生徒数が男子300人，女子200人であったとき，300とか200という数字は，人間にとっては男子生徒数と女子生徒数という意味のある数字である。これを情報という。ところで，A高校の生徒数の合計を電子計算機で計算させるとき，電子計算機はその数字の意味するところには関心がなく，ただ命令されたとおりに 300+200 の計算を行なうだけである。人間が数字の意味を考えずに，単に計算だけを行なう場合にも同じことがいえる。この場合の300とか200とかいう数字をデータという。情報は，人間が見てすぐにその意味がわかるように理解しやすい形で表わされることが望ましいが，データの方はむしろ計算などの処理に適した形で表わされていた方がつごうがよく，また，実際そういう形式で表わされている。

上の例では，数値の計算の場合を取り上げたが，データは数値の計算ばかりに使われるわけではない。この点も含めてデータという言葉を定義すると「事象，概念，命令などを表現するもので，人間もしくは機械による処理に適するように形式化されたもの」がデータということになる。これに対して情報とは，「一定の約束に基づいて人間がデータに割り当てた意味」のことをいうのである。なお，前述のプログラムも，広い意味でデータの中に含めて考えることが多いが，プログラムとデータとを区別していう場合もある。

3. 電子計算機におけるデータの表わし方　第1章では，卓上計算機の取り扱うデータ，すなわち，数字を2進数でどのように表わすかについて述べたが，ここでは，より一般的に文字や記号の表わし方について考える。電子計算機の中では，文字や記号も数字と同じように2進数で表わされる。

一例として，文字を2進6けた，すなわち6ビットで表わすことを考えてみよう。それには，数字の5は 000101，また，アルファベットのAは 100001 というように，あらかじめ約束しておけばよい。

---

(1) information

2進6けたでは，最小 000000 から，最大 111111 (10進表示で63) まで，64種類の数字，文字，記号を表わすことができる。したがって，0～9の数字10種，アルファベット26文字のほか，＋やーなど28種の記号を表わすことができる。同様に2進8けた，すなわち，8ビットで1字を表わすと，256 種類の数字，文字，記号を表わすことができるので，アルファベットのほか，カナ文字も表わすことができるし，記号の種類も多くとることができる。

数字を取り扱う場合には，上記のような表わし方では演算を行なうとき不便なので，第1章で説明した2進化10進法とか2進法とかで表わすことが多い。こうして，同じ2進符号表示であっても，それが2進法による数字を表わすのか，または2進化10進法による数字を表わすのか，あるいは上述の文字を表わしているのかによって，意味している内容が変わってくる。たとえば，%という記号は8ビット符号で 00100101 と表わされるが，これは2進化10進法では10進数の25となり，また2進数では10進数の37となる。00100101 を%と読むか，25と読むか，37と読むかは，あらかじめプログラムで決めておけばよい。たとえば，プログラムで2進化10進法で取り扱うことにしておくと，00100101 は10進で25という数字に解釈される。

このほか，プログラム自身も2進数で表わされるが，これについては2.6で述べることにする。

4. **ハードウェア**(1)**とソフトウェア**(2)　1で述べたように，電子計算機では機械そのものがあるだけではだめで，機械を働かすためのプログラムがなければその力を発揮することができない。もちろんその逆に，プログラムだけあっても機械がなければ，何も動かすことはできない。これはちょうど，卓上計算機が手もとにあっても，その使い方を知らなければ宝の持ちぐされであり，また逆に，計算する方法を知っていても計算機がなければ，実際に計算を行なうことができないのと同じである。

この機械に相当するものを**ハードウェア**といい，プログラム（卓上計算機の場合は，使用方法がこれに当たる）に相当するものを**ソフトウェア**という。

---

(1)　hardware　　(2)　software

2.1.2 電子計算機の構成と機能　第1章で述べた卓上計算機の構成と対比させて，電子計算機の構成を考えてみることにする（なお，第1章では入力回路とか演算回路のように回路という言葉を使ったが，電子計算機になると，これらの回路ははるかに大がかりになるので，**装置**(1)という言葉に昇格させることにする）。

電子計算機でも入力回路や入力変換回路に相当するものがあり，ここでは，これらをいっしょにして，**入力装置**(2)とよぶ。同様に，出力回路と出力変換回路とをまとめて**出力装置**(3)ということにする。

電子計算機においては**記憶装置**(4)はとくに大きくなり，主となる部分と補助となる部分に分けた方がつごうがよい。前者を**主記憶装置**(5)，また後者を**補助記憶装置**(6)という。

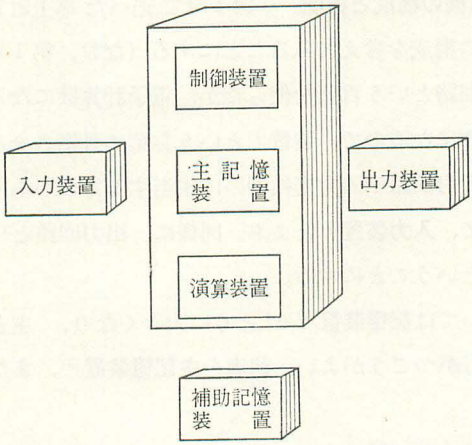
第1.2図の卓上計算機の構成図には示してないが，電子計算機の構成図を描く際に忘れてはならないものに**制御装置**(7)がある。卓上計算機の回路を説明する際，演算信号やけた送り信号を出すのに必要なものとして制御回路にふれたが（**1.5.4**参照），この制御回路は単に人間の操作どおりに制御信号を出していたのに対し，電子計算機の制御装置は，人間の操作を肩代わりして，すべての装置の制御をつかさどっており，いわば電子計算機のかなめとして，きわめて重要な役割りをになっているのである。

このほか，卓上計算機の演算回路に相当するものとして**演算装置**(8)がある。

以上のことを結論的にいうと，電子計算機を構成しているのは，**第2.1図**に示すように，つぎの装置である。

（1） 入力装置
（2） 出力装置
（3） 記憶装置（主記憶装置，補助記憶装置）
（4） 演算装置
（5） 制御装置

（1） unit　（2） input unit　（3） output unit　（4） memory
（5） main internal memory　（6） secondary memory, auxiliary memory
（7） control unit　（8） arithmetic unit



第2.1図　電子計算機の構成

である。

主記憶装置，演算装置および制御装置は，普通1組の箱に納められていて，これを**中央処理装置**(1)という。また，入力装置と出力装置を総称して**入出力装置**(2)という。さらに，入出力装置と補助記憶装置を含めて**周辺装置**(3)という。

これらをまとめて示すと，つぎのとおりなる。

- 電子計算機
  - 中央処理装置
    - 制御装置
    - 主記憶装置
    - 演算装置
  - 周辺装置
    - 補助記憶装置
    - 入出力装置
      - 入力装置
      - 出力装置

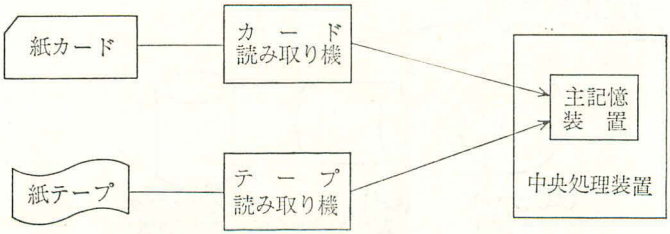
電子計算機で，これらの装置がどのような役目を果しているかを，つぎに簡単に説明することにする。

1.　入力装置　入力装置は，電子計算機へデータを送り込むために使われる装

(1)　central processing unit：略して CPU　(2)　input-output unit
(3)　peripheral equipment

置である。卓上計算機では，人間がボタンの操作で数値を送り込んでいたのに対し，電子計算機では，データは普通，紙のカードや紙のテープなどに記録されていて，それらを入力装置にかけると，その記録内容に応じて計算機内部での処理につごうのよい電気的符号が発生し，それが中央処理装置へ送られる（**第2.2図**参照）。

第2.1表には，入力されるデータの記録形式と，それに使用される入力装置の代表的な例が示されているが，このほかにも各種の入力装置が使用されている。



第2.2図　入力装置の働き

第2.1表　入力装置の種類

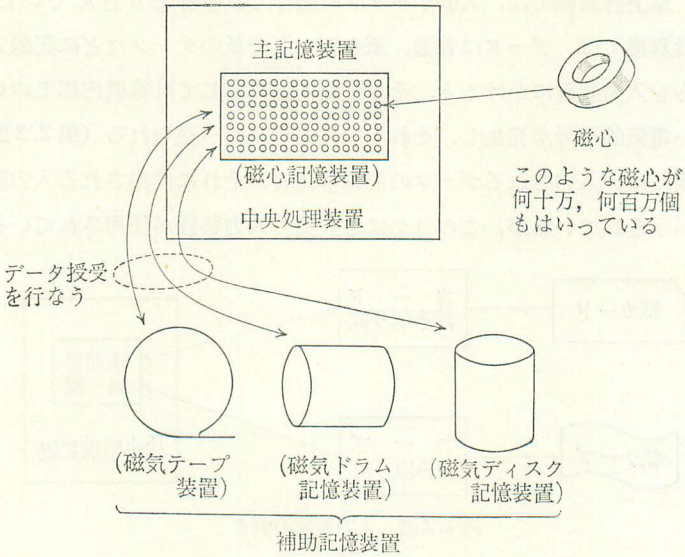
データの記録形式	入力装置
紙のカードにせん孔*	カード読み取り機
紙のテープにせん孔	テープ読み取り機

*　**せん孔**とは，カードや紙テープに穴をあけること。

2.　記憶装置　記憶装置は，入力されたデータ，出力されるデータ，中間結果など処理に必要なデータを記憶する装置であるが，卓上計算機では数字をレジスタという記憶回路に記憶していたのに対し，電子計算機の記憶装置では，数字以外に文字，あるいは各種記号を組み合わせて作られているデータや，さらにこれを処理するための手順（プログラム）まで記憶しておかなければならない。

ところで，レジスタは，電子計算機でもデータを一時的に記憶する記憶回路として使われているが，上に述べたようなぼう大なデータを記憶するためには，もっと別の装置が必要になる。このための装置を記憶装置という（データを一時的に記憶するレジスタについては，制御装置や演算装置の項で説明することにしよう）。

さて，電子計算機の記憶装置には，**第2.3図**に示すように主記憶装置と補助記憶



第2.3図　記憶装置のいろいろ

装置とがある。前者は中央処理装置の中にあって，当面の処理に必要なプログラムやデータを記憶する装置であり，また，後者は中央処理装置の外にあって，さしあたって必要がなく，主記憶装置に記憶しきれない大量のデータを記憶するために使われる装置である。

データを記憶するのに，どのような材料(媒体)を使用するかによって，第2.2表に示すような各種の補助記憶装置がある。

第2.2表　補助記憶装置の種類

データの記憶形式	補助記憶装置
磁気テープに記憶	磁気テープ装置
磁気ドラム（円筒）に記憶	磁気ドラム記憶装置
磁気ディスク（円板）に記憶	磁気ディスク記憶装置

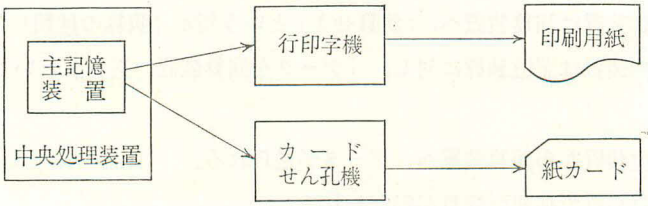
また，主記憶装置には，記憶材料として磁心を用いた磁心記憶装置が多く用いられている。以下，特に断わらない限り，記憶装置とは主記憶装置を指すことにする。

3.　演算装置　　演算装置は第1章で説明したような四則演算や，その他の演算

を行なう装置である。

4. **出力装置**　出力装置は，電子計算機で処理された結果を印刷などの方法により，外に出す装置である。卓上計算機が，計算結果を数字表示窓に表示すればよかったのに対し，電子計算機では，さまざまの出力方法がとられる。中央処理装置で処理された結果はその内容に応じた電気信号となって出力装置へ送られ，適当な符号に変換され，用紙に印刷されたり，紙カードにせん孔される（**第 2.4 図**参照）。

どのような形で出力するかによって，**第 2.3 表**のような各種の出力装置が用いられる。



第 2.4 図　出力装置の働き

第 2.3 表　出力装置の種類

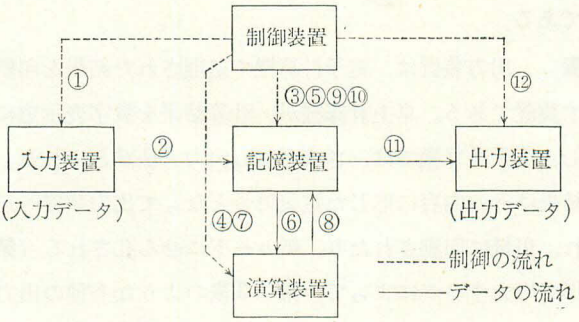
データの出力形式	出　力　装　置
用紙に印刷	行印字機
紙カードにせん孔	カードせん孔機
紙テープにせん孔	テープせん孔機

5. **制御装置**　電子計算機の各装置の動作は，制御装置によって制御される。入力データを読み，それを記憶，演算し，演算結果を出力するという処理を行なう場合，それに必要な制御がどのように行なわれるかを，**第 2.5 図**により簡単に説明する（以下，説明の番号は，図中各部の番号を表わす）。

① 制御装置から入力装置に，「データを入力せよ」という指示が与えられる。

② 入力装置は，データを読んで，それを記憶装置へ送る。

③ 制御装置は記憶装置に対して，「データを記憶せよ」という指示を出す。記憶装置は，入力されたデータを記憶する。



○内番号は説明文の番号と一致している。

第2.5図　データの処理と制御の仕方

④　制御装置は演算装置へ，「演算せよ」という指示（演算の種類）を与える。

⑤　制御装置は記憶装置に対し，「データを演算装置へ送れ」という指示を出す。

⑥　記憶装置から演算装置へ，データが送られる。
演算装置で必要な演算が行なわれる。

⑦　制御装置は演算装置に対し，「計算結果を記憶装置へ送れ」と指示する。

⑧　演算装置から記憶装置へ結果が送られる。

⑨　制御装置は記憶装置に対して，「結果を記憶せよ」という指示を出す。
結果が記憶装置に記憶される。

⑩　制御装置から記憶装置に，「結果を出力装置へ送れ」と指示する。

⑪　記憶装置から出力装置へ，結果のデータが送られる。

⑫　制御装置は出力装置に対し，「結果のデータを出力せよ」と指示する。
出力装置で結果が出力される。

これで，1つのデータ処理が終了したので，続いて，つぎのデータ処理にかかる。

このように，制御装置の指示に基づいて，各装置が整然と必要な動作を行ない，処理を進めていく。とはいっても，制御装置が勝手にこのような処理手順を考え，実行を制御しているのではない。前にも述べたように，データを処理するのに必要な手順，すなわち，プログラムを人間が作って，それを前もって電子計算機に与え

ておくことが必要なのであって，これがあってはじめて制御装置は与えられた手順に従って，必要な制御を行なっていくのである。

ここで，卓上計算機と電子計算機の機能的な比較をまとめて第2.4表にかかげておく。

第2.4表 比 較 表

	卓上計算機	電子計算機
入力の仕方	人間が数値，計算種別のボタンを押す。	各種入力装置を使って，データを入力する。
出力の仕方	表示窓に現われた結果を人間がみて，転記する。	各種出力装置を使って，データを出力する。
記憶	入力された数値などを，レジスタで記憶する。	データ処理の手順，各種データを主記憶装置，補助記憶装置で記憶する。
演算	四則演算ができる。	四則演算のほか，論理演算* もできる。
制御	計算処理の手順は，人間が覚えていて，逐次，人間が制御を行なう。	与えられた処理手順を電子計算機が記憶していて，それに従って，制御装置が必要な制御を行なう。

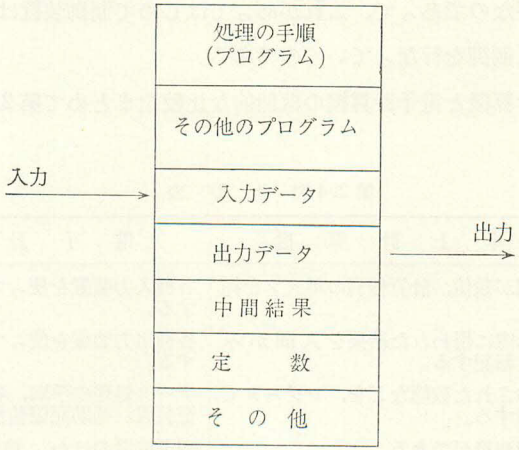
* 論理演算とは，比較，判断などを指す。これについては，後で述べる。

2.2 主記憶装置

2.2.1 概 要

第1章で述べたように，卓上計算機にはフリップフロップ回路などを用いたレジスタというものがあって，入力された数字，最終結果，あるいは中間結果などを一時的に記憶するのに用いられている。たとえば，5＋3を計算する場合には，入力された5および3の数値，ならびに結果の8という数値をレジスタに記憶する。このように，卓上計算機も記憶という機能は備えているが，その記憶装置は，たかだか2，3種類の数値を記憶するだけのごく簡単な，小規模のものである。

ところが，電子計算機の記憶装置になると，入力されたデータ，中間結果，最終結果などを記憶することはもちろん，データの処理手順（すなわち，プログラム），さらには処理な必要な定数などを記憶するだけの大容量のものが必要になってくる（第2.6図参照）。すなわち，電子計算機では，大量のデータを高速で処理することが要求されるので，これらデータ等を記憶する記憶装置は，卓上計算機のそれとは



第2.6図　主記憶装置の記憶内容

比較にならないほど，高性能で大規模のものが必要である。

前述のように，電子計算機の記憶装置には主記憶装置と補助記憶装置とがあるが，本節ではまず，代表的な主記憶装置である磁心記憶装置について，その記憶の原理，構成および動作について説明することにする。

2.2.2　磁心による記憶の原理

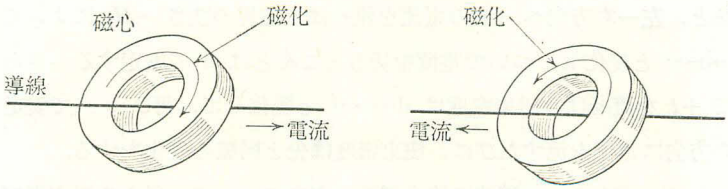
1.　磁　心　　電子計算機の主記憶装置として最も広く使用されているのは，**磁心記憶装置**(1)である。磁心記憶装置で使われている記憶素子は，フェライト(2)の小さなドーナツ状鉄心で，これを**磁心**(3)または**磁気コア**(3)という。

第2.7図は磁心の形状を示したもので，実用化当時は直径 1.25 mm 程度であったが，だんだん小形化され，現在では 0.5 mm とか，さらには 0.3 mm などという小形の磁心まで作られるようになった。

2.　記憶の原理　　第2.7図 (**a**) のように，磁心の中を通した導線に，左→右方向に電流を流すと磁心は矢印の方向に磁化され，電流を切っても磁化された状態を持続する。つぎに，同図 (**b**) のように右→左方向に電流を流すと，磁心は，こんど

(1)　magnetic core memory

(2)　ferrite：フェライトは酸化鉄を主とした酸化金属粉を焼結したもので，永久磁石の材料として広く使用されている　　(3)　magnetic core



(a) “1”が記憶されている　　(b) “0”が記憶されている

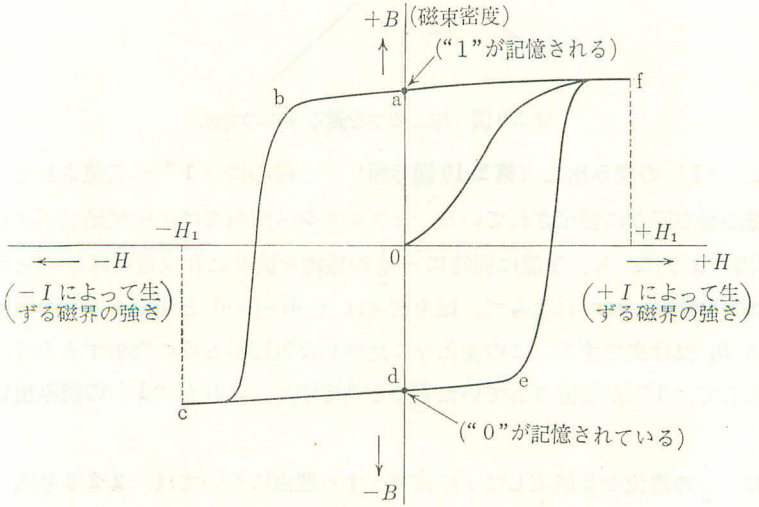
第2.7図　磁心の記憶

は逆方向に磁化され，再び電流を切ってもその方向に磁化された状態を持続する。

いま，図(a)の状態を“1”が記憶されていると約束し，図(b)の状態を“0”が記憶されていると約束すれば，1個の磁心は1ビットのデータを記憶する記憶素子とみなすことができる。

磁心が磁化される様子を，**第2.8図**のヒステリシス特性曲線で調べてみよう。

磁心の中を通っている導線に左→右方向へ $+I$ の電流を流せば，それにより生ずる磁界の強さ $+H_1$ により，磁束密度 B は 0→f に達し，そのあと電流を切っても磁束密度は0にもどらず，a で安定する。

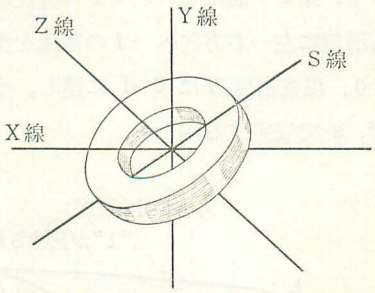


第2.8図　ヒステリシス特性曲線

そのあと，左→右方向へ $-I$ の電流を流せば，磁界の強さ $-H_1$ によって磁束密度は a→b→c と変化し，ついで電流を切るとこんどは d で安定する。さらに，この状態で $+I$ を流せば，磁束密度は d→e→f と変化して，再び a 点で安定する。以下逆の方向に電流を流すたびに，磁束密度は先と同様の変化をする。

このことは，磁心の中の導線に流す電流の方向によって，磁心の磁束密度の状態を制御できることを意味しており，磁束密度がa点のときを“1”が記憶されている状態とし，また，d点にあるとき“0”をが記憶されていると約束すれば，前述のように1個の磁心で“1”か“0”か，すなわち，1ビットのデータを記憶することができることになる。

3.　読み出し(1)と書き込み(2)　　**第2.9図**のように，磁心の中にX，Y，ZおよびS(3)の4本の導線を通したものを考える。

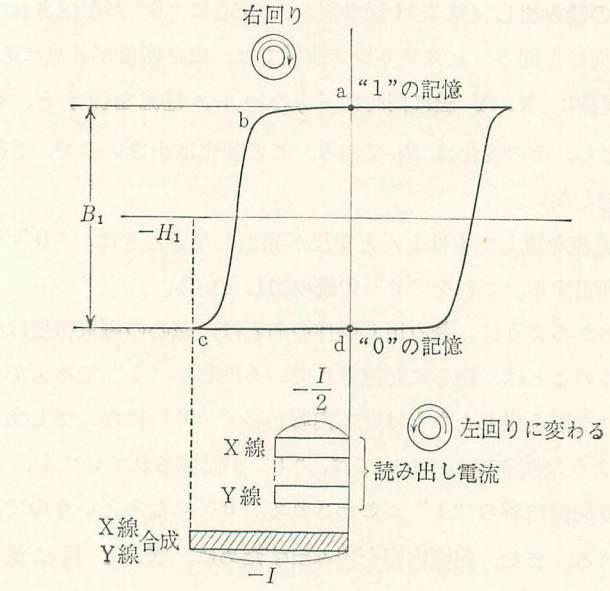


第2.9図　磁心の中を通る4本の導線

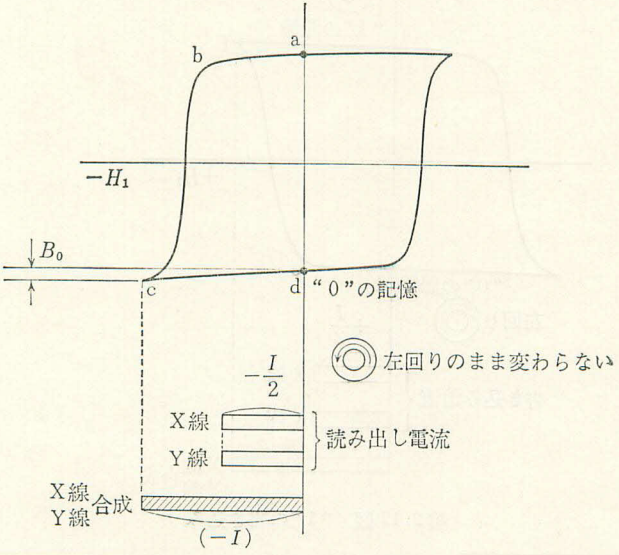
a.　“1”の読み出し（第2.10図参照）　　磁心に“1”が記憶されているとき（磁心が右回りに磁化されている。ヒステリシス曲線では，磁束密度がa点にある），図のように，X，Y線に同時に $-\frac{I}{2}$ の電流を流せば合成電流は $-I$ となり，これによる磁界 $-H_1$ によって，磁束密度は a→b→c→d と変化し，その間に磁束密度は B_1 だけ変化する。この変化分に比例した電圧がS線に発生するので，これを検出して“1”が記憶されていたものと判定する。これを**“1”の読み出し**という。

なお，$\frac{I}{2}$ の電流を2個流して I の電流とする理由については，**2.2.3**参照

（1）　to read (out)　　（2）to write　　（3）　S線をR線ということもある



第2.10図 “1” の読み出し

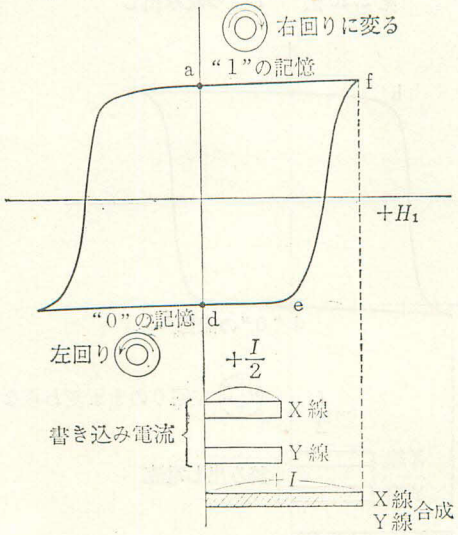


第2.11図 “0” の読み出し

b.　“0”の読み出し（第2.11図参照）　磁心に“0”が記憶されているとき（磁心の磁化方向は左回り，ヒステリシス曲線では，磁束密度がd点にある），“1”の読み出しと同様に X，Y 線に同時に$-\frac{I}{2}$のパルス電流を流すと，磁束密度はd→c→d と変化し，その変化は B_0 である。この変化は小さいので，S線にはほとんど電圧は誘起しない。

こうして，電流を流してもほとんど電圧が発生しないときは，“0”が記憶されていたものと判定する。これを“0”**の読み出し**という。

上の2例でわかるように，読み出し動作のあとは，磁心の磁束密度は必ずd点になっている。このことは，磁心に記憶されている内容が“1”であっても“0”であっても，それを読み出したあとは記憶内容が必ず“0”になってしまうことを意味する。このような読み出しにおいては，“1”が記憶されていても，それを読み出すことにより記憶内容の“1”がこわされて“0”になるというので，**破壊読み出し**(1)といわれる。また，記憶内容を読み出すために，X，Y 線に流す電流を**読み出し電流**(2)という。



第2.12図　“1”の書き込み

（1）　destructive reading　　（2）　read current

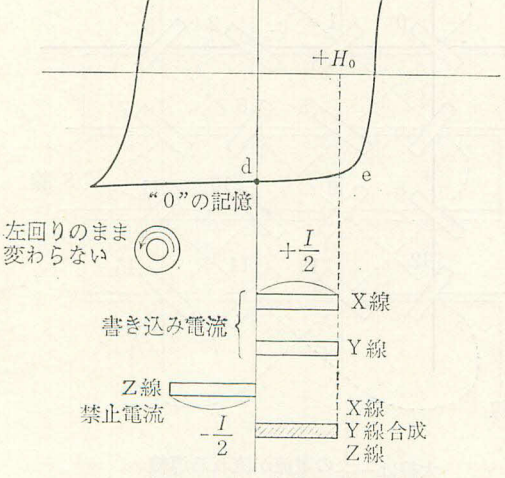
c. “1”の書き込み（第2.12図参照） 磁心記憶装置に書き込む動作は，必ず読み出し動作のあとに行なわれる。ということは，磁心の記憶内容が“0”，すなわち，磁束密度がd点の状態において書き込み動作が開始されることを意味する。

X，Y線に同時に $+\frac{I}{2}$ の瞬間電流を流すと，その合成電流 $+I$ になり，磁束密度は d→e→f と変化しa点で安定する。これが**“1”の書き込み**である。

d. “0”の書き込み（第2.13図参照） “0”の書き込みを行なう場合は，X，Y線に $+\frac{I}{2}$ の瞬間電流を流すことは前と同様であるが，これと同時にZ線に $-\frac{I}{2}$ の瞬間電流を流す。その結果，X，Y およびZ線による合成電流は，

$$\left(+I\right)+\left(-\frac{I}{2}\right)=+\frac{I}{2}$$

となり，そのため磁束密度はd→eと変化するだけで再びd点にもどって安定し，“0”が記憶されたままの状態を継続する。これが**“0”の書き込み**である。



第2.13図 “0”の書き込み

e. 再書き込み 破壊読み出しと称する読み出し（**2. 3. 2の1** 参照）の場合は，読み出しのたびに記憶内容が破壊されて“0”になってしまうので，記憶内容を復元しておかねばならない。すなわち，読み出した内容を別のレジスタにたくわ

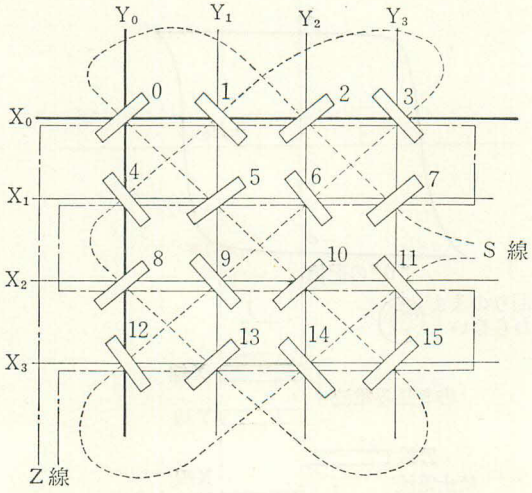
えておき，つぎの時点で読み出した内容を書き込んでおく必要がある。これを**再書き込み**という。

書き込み，あるいは再書き込みのとき，X, Y 線に流す電流を**書き込み電流**(1)といい，また，“0”を書き込むときZ線に流す電流を“1”の書き込みを禁止するという意味で，**禁止電流**(2)という。

2.2.3　電流一致方式

いままでは，磁心がただ1個の場合の読み書きの動作の説明であったが，実際の磁心記憶装置には非常に多くの磁心が組み込まれているので，この中から読み書きの対象となる1個の磁心を選び出して読み書きを行なう必要がある。

一例として，**第2.14図**のように縦横4個ずつ並べられた0番～15番の磁心の中から，0番の磁心を選択して読み書きを行なう方法を説明する。



太線は$-\frac{I}{2}$の電流が流れる導線

第2.14図　0番磁心の選択

図のように，整数と並べられた磁心の横方向に X_0～X_3，また，縦方向に Y_0～Y_3

(1) write current　(2) inhibit current

の導線を通す。一方，Z線，S線はすべての磁心の中を通しておく。

いま，X_0，Y_0 に同時にそれぞれ $-\frac{I}{2}$ の電流を流すと，

X_0，Y_0 の合成電流 $-I$ が流れる磁心……0番

X_0 あるいは Y_0 の $-\frac{I}{2}$ が流れる磁心……1，2，3，4，8，12番

まったく電流が流れない磁心……5，6，7，9，10，11，13，14，15番

となる。ここで，$-\frac{I}{2}$の電流では第2.10図のヒステリシス曲線においてa～bであるようにしておけば，0番磁心に“1”が記憶されているときに限り電圧が発生することになる。これは0番磁心だけが読み書きの対象として選ばれたことを意味する。このような方法により特定の磁心を選択する方式を，**電流一致方式**という。

ここで，磁心に対する読み出し，書き込みに，電流一致方式を採用している理由を考えてみる。

もし，電流一致方式を採用しなければ，4行4列の合計16個の磁心にそれぞれ導線を通して，その電流を制御しなければならない。しかし，電流一致方式では縦横4本ずつ合計8本のX，Y線を必要とするだけで，導線の所要数は半分ですむ。

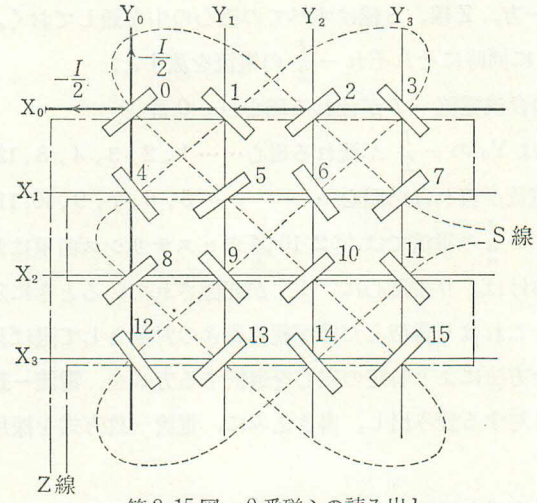
普通，磁心記憶装置は，非常に多くの磁心を平板状に並べたものを使用している(詳細は後述する)。

たとえば，4096個の磁心を平板状に並べると，縦横64個ずつ並ぶことになり，($64\times64=4096$)，また，16384個の場合は，縦横に128個ずつ並ぶことになる($128\times128=16384$)。このような多数の磁心を制御するのに，もし，電流一致方式をとらないときは，上に述べたように，磁心の数だけ，すなわち，前者は4096本，後者は16384本の導線を必要とするのに対し，電流一致方式では，前者で$64\times2=128$本，後者で$128\times2=256$本の導線により目的を達することができる。

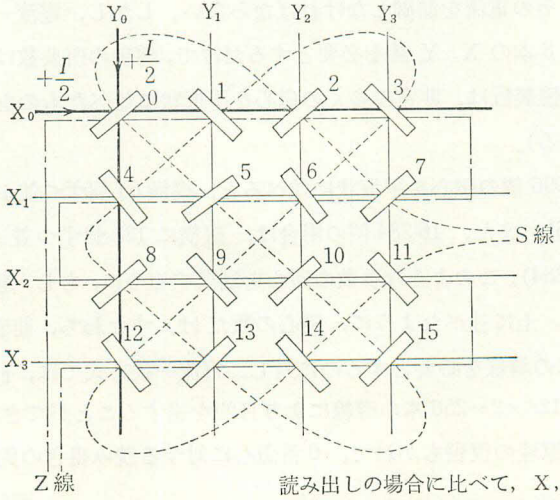
ここで，記憶原理の復習もかねて，0番磁心に対する読み書きの動作を整理しておく。

1. 読み出し　第2.15図に示すように，X_0，Y_0 に同時に $-\frac{I}{2}$ の瞬間電流を流すと，0番磁心だけに合成電流 $-I$ が流れるので，0番磁心の記憶内容が“1”であれば，S線に誘起電圧が発生するし，“0”であれば発生しない。

読み出したあと，0番磁心の内容は“0”になる。



第2.15図　0番磁心の読み出し



第2.16図　0番磁心に対する“1”の書き込み（再書き込みも同じ）

2.　書き込みあるいは再書き込みで“1”を書き込むとき　第2.16図に示すように，X_0，Y_0 に同時に $+\frac{I}{2}$ の瞬間電流を流すと，0番磁心に合成電流 $+I$ が流れて“1”が書き込まれる。

3. 書き込みあるいは再書き込みで“0”を書き込むとき　第2.17図に示すように，X_0，Y_0 に $+\frac{I}{2}$，Z線に $-\frac{I}{2}$ の瞬間電流を同時に流すと，0番磁心の合成電流は $+\frac{I}{2}$ となり，“1”の書き込みが禁止され，“0”が記憶されたままとなる。

以上の説明でわかるように，磁心を使っての記憶には，つぎの重要な性質がある。

（1） 記憶には読み出し動作と，それに引き続く書き込み動作の2つのサイクルがある。

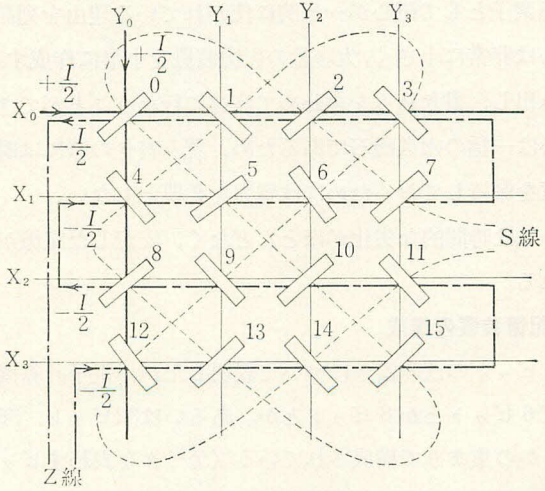
読み出しのとき

読み出し動作	再書き込み
（読み出した内容を使う）	（記憶内容を復元する）

書き込みのとき

読み出し動作	書き込み
（読み出した内容は使わない）	（新しい内容を書き込む）

（2） 記憶内容を何度読み出しても，その内容は変わらない。なぜなら，再書



合成電流が0となる磁心……1，2，3，4，8，12
合成電流が $+I/2$ となる磁心……0
$-I/2$ が流れる磁心……5，6，7，8，9，10，11，13，14，15

第2.17図　0番磁心に対する“0”の書き込み（再書き込みも同じ）

き込みで復元するから。

（3）　しかし，新らしい内容を書き込めば，前に記憶していた内容は失われて，新らしいものに変わる。

記憶装置に読み出しを指示してから，記憶装置で読み出し動作と再書き込み動作が行なわれ，つづいて，つぎの読み出しのための準備が完了して，読み出しの指示が受け付け可能となるまでの時間を**サイクル・タイム**(1)という。通常の磁心記憶装置のサイクル・タイムは，1～数 μs（マイクロ秒＝10^{-6} s）であるが，最近では，1 μs 以下で読み書きできる磁心記憶装置も使われている。

また，読み出しの動作を指示されてから，要求のあった記憶内容が取り出されるまでの時間を**アクセス・タイム**(2)，あるいは**呼び出し時間**(2)といい，サイクル・タイムのほぼ半分程度である。

サイクル・タイムやアクセス・タイムは，電子計算機の性能を知る目安となるもので，電子計算機のカタログには必ず記載されている。

以上，磁心による記憶の原理，方法について述べたが，最後に，電子計算機の主記憶装置の記憶素子として磁心が一般的に使われている理由を列記すると，

（1）　磁心は非常に小さく，大容量の記憶装置を小形に作成することができる。

（2）　読み出し，書き込みをきわめて高速に行なうことができる。

（3）　磁心は一種の永久磁石であるため，読み書きの際には瞬間電流を必要とするが，記憶を保持しておくためには電力を必要としない。

（4）　磁心には時間的な劣化がほとんどなく，安定した記憶ができる。

などがあげられる。

2.2.4　磁心記憶装置の構成

前項では，1ビットの読み書きについて説明したが，電子計算機で取り扱うデータは，たとえば6ビットとか8ビットとか，あるいは32ビット，36ビットというように，何ビットかの集まりで構成されている（データを表わすビットのほかにも，制御やチェックのためのビットも必要である）。

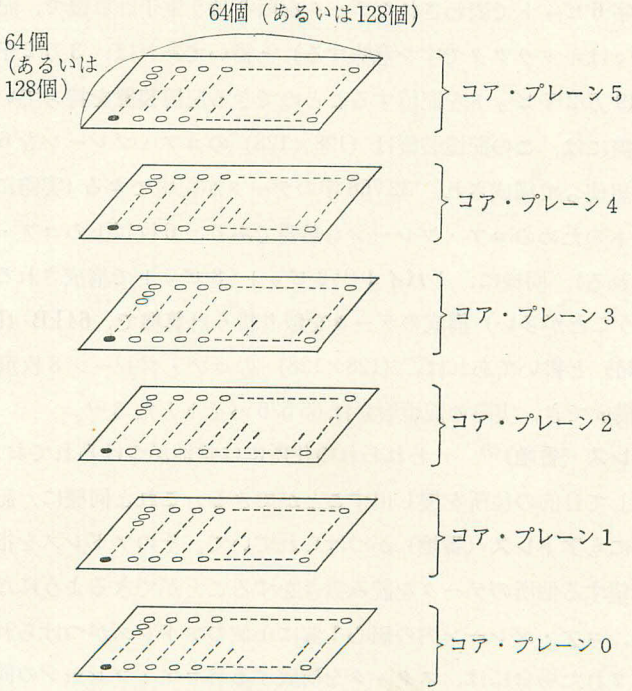
そこで本項では，複数ビットで構成されているデータを記憶する方法について説

（1）　cycle time　　（2）　access time

明する。

1. 磁心マトリクス(1)とスタック(2)　実際の磁心記憶装置は，たとえば**第2.18図**のように，縦横64個ずつ，64×64＝4 096個の磁心を平板状に並べたものや，縦横128個ずつ，128×128＝16 384個を並べたものを単位として用いるのが普通である。このように構成された磁心の集まりを**磁心マトリクス**といい，また，1枚の磁心マトリクスのことを**コア・プレーン**(3)ともいう。

いま，このようなコア・プレーンを6枚積み重ねると，上から下へとくし刺しになった6個の磁心の組ができる。この1組の磁心をまとめて磁化させれば，たとえ



1字6ビットの記憶は，黒丸の磁心で行なわれる。
1字6ビットで，64×64＝4 096 字分記憶できる。
もし，128個ずつ並べると，16 384字分記憶できる。

第2.18図　ス　タ　ッ　ク

(1)　core matrix　　(2)　stack　　(3)　core plane

ば1字6ビットで構成されるデータ1字分を記憶させることができる。

このように，コア・プレーンを必要枚数だけ積み重ねて記憶装置の単位にしたものを**スタック**という。

前述のコア・プレーンを重ねたスタックでは，前者が4096字分，後者では16384字分のデータを記憶できることになる。

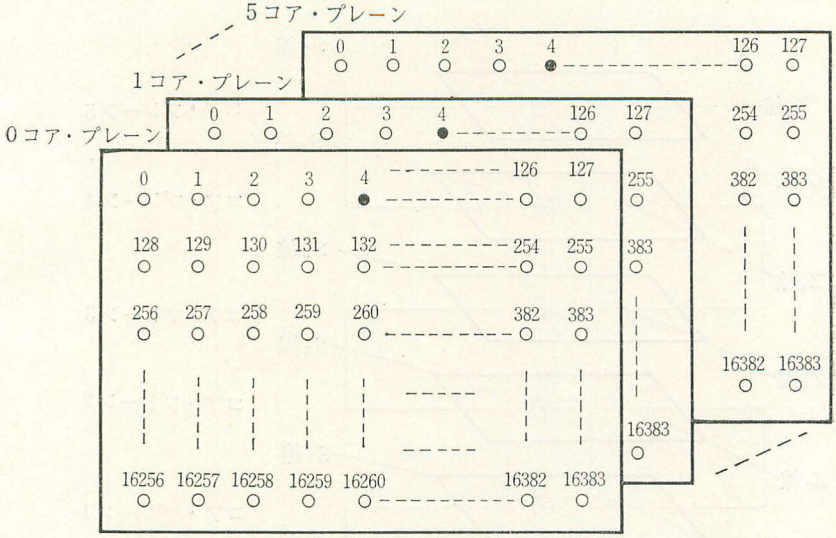
電子計算機のカタログには，たとえば記憶容量32 kcとか，64 kBなどの表示があるが，これは，その電子計算機の記憶装置にどれだけのデータが記憶できるかを示している。

いま，1字6ビットで表わされたデータを取り扱う電子計算機で，記憶容量32kc（kは10^3，cはキャラクタで字を意味する）と書いてあれば，3万2千字のデータすなわち，19万2千ビットを記憶することのできる記憶装置を持っていることを意味する。実際には，この記憶装置は（128×128）のコア・プレーンを6枚重ねたスタックを2組使って構成され，32768字のデータが記憶できる（実際には，制御用などのビットのためのコア・プレーンも必要であり，6枚以上のコア・プレーンがスタックされる）。同様に，1**バイト**(1) 8ビット（8ビットで構成される符号列を1バイトということが多い）構成のデータを取り扱う計算機で，64 kB（kは10^3，Bはバイトの略）と書いてあれば，（128×128）のコア・プレーン8枚重ねたものを4組使って構成でき，実際の記憶容量は65536バイトとなる(2)。

2. アドレス（番地）(3)　われわれの住所には番地がつけられており，その番地を索引として目的の住所を捜し出すことができる。これと同様に，記憶装置の中の記憶場所にも**アドレス**（番地）がつけられていて，そのアドレスを指定することにより，希望する個所のデータを読み書きをすることができるようになっている。

すなわち，コア・プレーン内の磁心1個に1個のアドレスがつけられ，これがスタックに組まれた場合には，スタックを構成する各コア・プレーンの同じ個所には同一アドレスがつけられる。したがって，あるアドレスを指定すると，スタックされた各プレーンの同1個所にある1組の磁心が指定される。

（1）byte　（2）電子計算機の容量を示す場合，kは1024を意味するのが普通である。したがって，64 kBは64×1024=65536 Bの容量となる。　（3）address



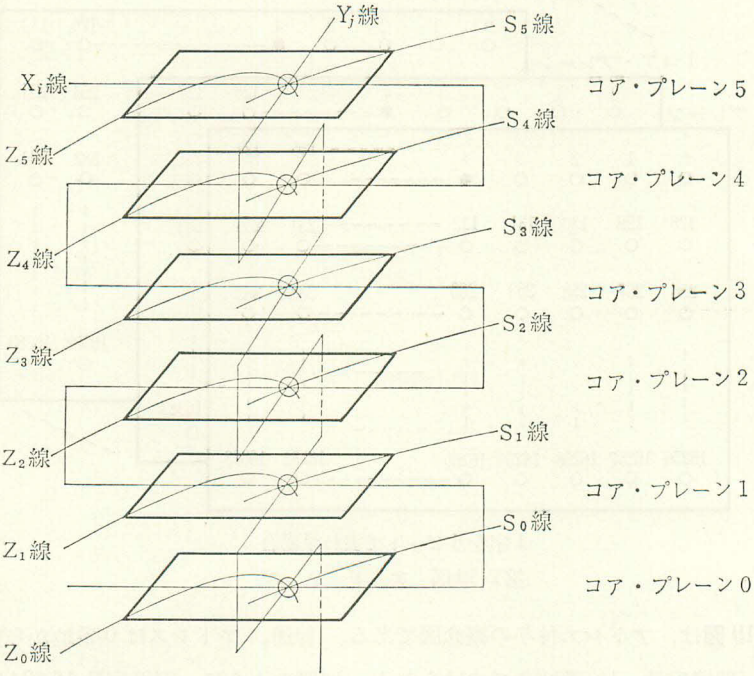
1字を6ビットで表わす場合

第2.19図 アドレス

第2.19図は，アドレス付与の概念図である。普通，アドレスは0番地から始まって，（記憶容量−1）番地までつけられる。同図のように，記憶容量16 384字のものであれば，0番地から16 383番地まで付与されることになる。

第2.19図の例において，記憶装置の中からデータの読み出し，あるいは書き込みを行ないたい特定のアドレスを選択するのには，たとえば，データの読み書きしたいアドレスが4番（図の黒丸）の場合，1行目の5列目というように指定すれば，0～5のコア・プレーンの4番地にあるすべての磁心が指定される。この指定の方法は，電流一致方式による磁心の指定とまったく同じ方法である。

3. スタックでの記憶　**第2.20図**は，スタック内のX，Y，ZおよびS線の配線の様子を示したものである。X，Y線は，各プレーン共通に，同じ行，あるいは列の磁心の中を通っている（詳細は，前出の第2.14図を参照）。第2.20図では，X_i，Y_jでその配線の様子を示している。また，Z線，S線は，コア・プレーンごとに設けられていて，1枚のコア・プレーン内のすべての磁心の中を通っている（第2.20図では，簡単のためコア・プレーンあたり1個の磁心についてだけ描い

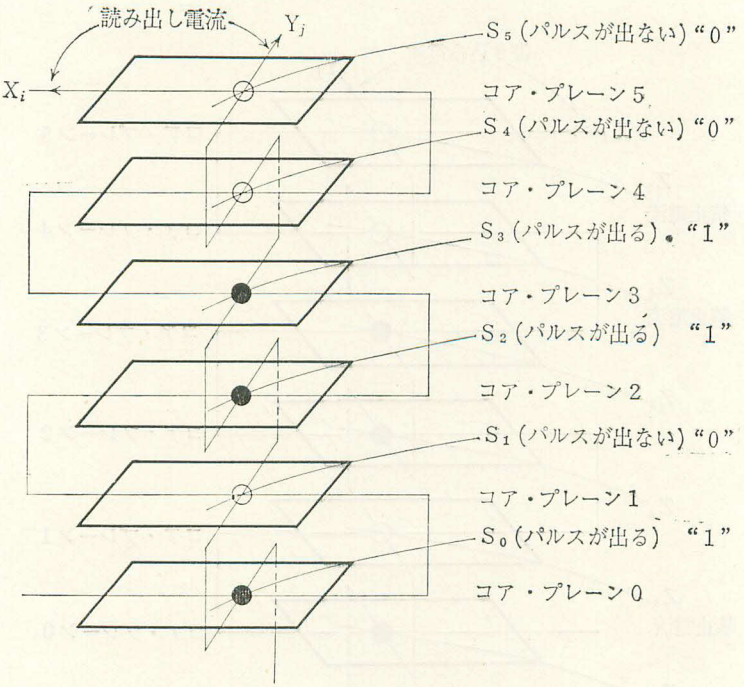


第2.20図　X，Y，ZおよびS線の配線

ている)。

第2.21図において，コア・プレーン0から上へ1，2と順に101100の2進データが記憶されている場合の読み取り，書き込みの動作は，つぎのように行なわれる(図で●印は“1”，○印は“0”が記憶されているものとする)。すなわち，読み出したいデータが記憶されているアドレスが指定され，それに対するX_i，Y_j線を選択して，それぞれに読み出し電流を流すと，“1”が記憶されている。0，2および3のコア・プレーンのS線には誘起電圧が発生し，その他のS線には発生しないので，101100のデータを読み出すことができる。

この場合の読み出しは，破壊読み出しであり，すべての磁心は“0”になってしまうので，再書き込みを行なう。そのためには，いま読み出したデータをもう一度書き込めばよいので(読み出した内容は，別にレジスタに一時的にたくわえてお

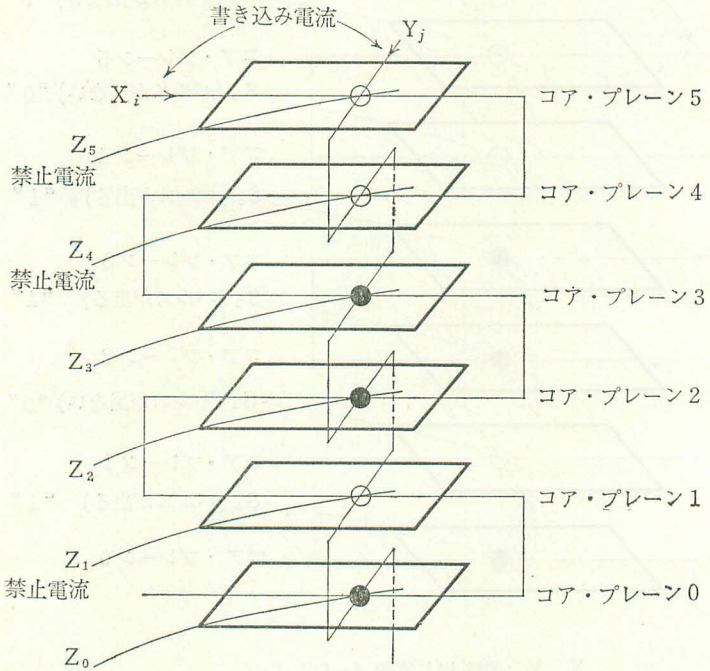


X_i, Y_j：読み出し電流（$-I/2$ ずつ）
S_0～S_5：記憶内容が"1"のとき電圧が出る。
いま，S_0, S_2, S_3 に電圧が出る。Z線は省略
第2.21図 読 み 出 し

く），**第2.22図**のように，書き込み電流を X_i, Y_j 線に流すと同時に，"0"のままにしておきたいコア・プレーン1，4，5のZ線に禁止電流を流すと，読み出したと同じデータが同じアドレスに再書き込みされる。

なお，一般の書き込みのときは，読み出した内容を捨てて，新らしいデータを上と同様の手順で書き込めばよい。

4. **検査の方法**　磁心記憶装置に記憶されるデータは，1字ごと，または1バイト，あるいは何字かの組ごとに，誤りの有無が検査される。検査の方法は，データを表わす2進符号に検査用のビットを付加し，符号中の"1"のビットの数の合計が奇数個（あるいは偶数個）になるように，検査用のビットを"1"，あるいは"0"にして記憶しておき，読み出しに際して，1字ごとに"1"のビットの数を数

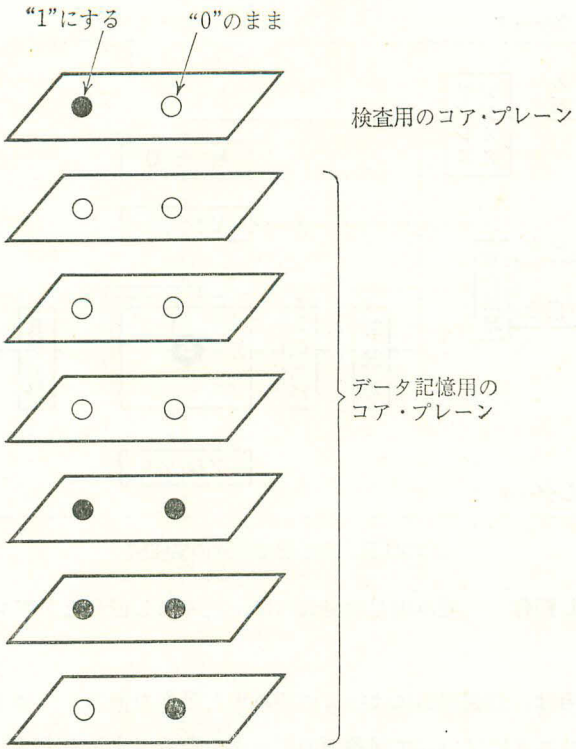


X_i, Y_j：書き込み電流（$+I/2$ ずつ）
Z_0～Z_5：“0”を書き込むとき禁止電流を流す。
いまは Z_1, Z_4, Z_5 に流す。S線は省略
第2.22図　再書き込み（あるいは書き込み）の仕方

え，それが正しく奇数個(あるいは偶数個)あるかどうかを調べる方法がとられる。これを**奇偶検査**(1)または**パリティ・チェック**(1)といい，奇数個を基準にする**奇数パリティ・チェック**(2)と，偶数個を基準にする**偶数パリティ・チェック**(3)とがある。

磁心記憶装置は，検査用のビット（これを**パリティ・ビット**ともいう）のためのコア・プレーンを持っている。たとえば，奇数パリティ・チェックを採用している場合は，**第2.23図**のように，データを構成する“1”のビット数が偶数ならばパリティ・ビットを“1”とし，データの“1”のビット数が奇数のときはパリティ・ビットを“0”のままにしておく。

（1）parity check　（2）odd parity check　（3）even parity check



• 印は“1”になっているビットを示す。
• 印が奇数個あることに注意

第2.23図 奇偶検査（奇数パリティチェックの例）

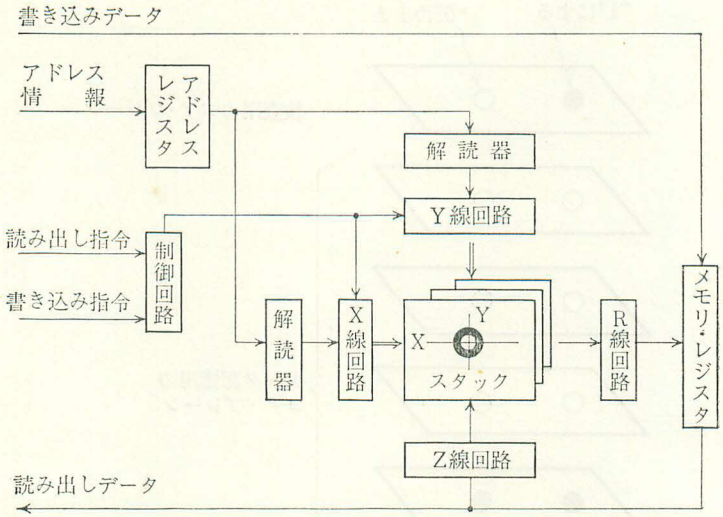
2.2.5 磁心記憶装置の動作

第2.24図は，磁心記憶装置の構成図である。

アドレス・レジスタ(1)は，記憶装置のアドレスを指定する**アドレス情報**がたくわえられるレジスタであり，**メモリ・レジスタ**は，読み出しデータや書き込みデータがたくわえられるレジスタである。また，解読器は，アドレス情報をもとにして，X，Y線を選択するための回路である。

X，YおよびZ線に流す電流は，それぞれ，X線回路，Y線回路およびZ線回路で制御される。

(1) address register



第2.24図　磁心記憶装置の構成図

1.　読み出し動作　読み出しの場合には，読み出し指令とアドレス情報が送られてくる。

読み出し指令は，制御回路にはいって読み出し動作を指示し，アドレス情報は，アドレス・レジスタにはいって何番地のデータを読み出すかを指定する。解読器は，アドレス情報をもとにして該当するX線，Y線を選択し，X線回路，Y線回路に通知する。

一方，制御回路から，X線回路，Y線回路に対して読み出し指令が出されると，読み出し動作が行なわれ，読み出されたデータはメモリ・レジスタにたくわえられたのち，データを使用する装置へその内容が送り出される。

続いて，再書き込み動作にはいる。読み出しに使ったX，Y線，およびメモリ・レジスタの内容（読み出されたデータがたくわえられている）によりZ線を制御することにより，読み出しを行なったアドレスにデータが再書き込みされる。

2.　書き込み動作　この場合は，書き込み指令，アドレス情報および書き込みデータが送られてくる。

書き込み指令およびアドレス情報により，まず前と同様の読み出し動作が行な

われる。読み出したデータがメモリ・レジスタにたくわえられた時点で，書き込みのデータがメモリ・レジスタへ送られ，メモリ・レジスタの内容を新らしい書き込みのデータに書き換える。

続いて書き込み動作にはいり，再書き込みと同様の方法で，メモリ・レジスタの内容が指定アドレスへ書き込まれる。

2.3 補助記憶装置

2.3.1 概 要

大量のデータ処理を高速に行なうためには，主記憶装置の記憶容量はできるだけ大きいことが望ましいが，主記憶装置の価格は高いため，いたずらに主記憶装置の容量を増大するのは経済的でない。また，実際のデータ処理においても，常に大容量の主記憶装置をかかえこんでおく必要のない場合もある。

たとえば，給与計算をする場合を考えると，机の上に置いてある分厚い給与台帳の中身を，全部主記憶装置に記憶させておく必要はなく，これから計算しようとする人の分のデータを記憶するというように，台帳の中の大量のデータのうちから必要なデータだけを取り出して，それに対する計算を行なうことで足りる。

このように，大量のデータを別に記憶しておき，必要なつど，必要なデータを取り出して主記憶装置に記憶しなおすことにより，主記憶装置の容量を補うための装置が，補助記憶装置である。

補助記憶装置に要求される条件は，

（1） 大量のデータが記憶できること。

（2） データの記録および取り出しが高速で行なえること。

（3） 単位データ当たり（すなわち，ビット当たり）の価格が安いこと。

などである。

現在用いられている代表的な補助記憶装置には，

磁気テープ装置，　　磁気ドラム記憶装置，　　磁気ディスク記憶装置

などがある。

2.3.2 磁性面上におけるデータの記録

1. 書き込みと読み出しの原理　家庭用のテープ・レコーダを考える。一方のリールに巻かれている磁気テープが，他方のリールに巻き取られる途中に，消去，録音および再生用ヘッドが設けられている。

音を録音する際には，録音電流を録音ヘッドに流すことにより，ヘッドの表面を走っているテープの表面を磁化して録音が行なわれる。また，録音された音を再生する際には，録音ずみのテープが再生ヘッドの表面を走るときに，テープ上の磁化に応じて発生する電流を増幅して行なわれる。こうして，録音も再生も，磁気的な方法で行なわれている。

電子計算機の補助記憶装置として，広く用いられている磁気テープ装置，磁気ドラム記憶装置，磁気ディスク記憶装置なども，その名が示すように，磁気的な方法でデータを記憶する装置である。

磁気テープのように，データを磁気的に記録しておくものを**磁気記録媒体**というが，これには，磁性テープのように磁気材料を塗布したプラスチック製テープのほか，金属円筒や金属円板などの表面に鉄酸化物の微粒子を薄く塗ったり，ニッケル・コバルトのような強磁性材料をめっきしたものが用いられる。

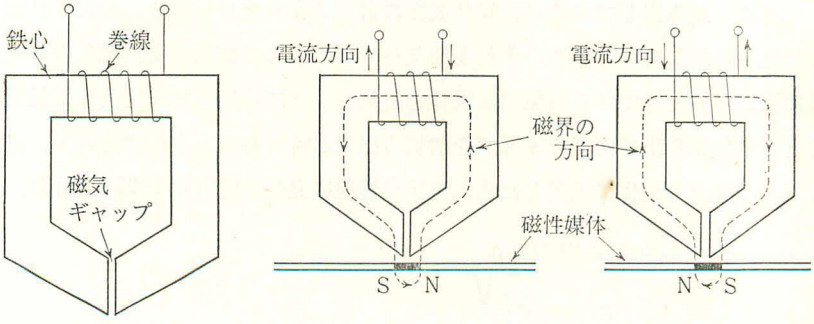
こうした記録媒体の表面を**磁気ヘッド**(1)によって磁化してデータを記録したり，逆に，磁気ヘッドにより媒体上の磁化の状態を検出して，データの再生が行なわれる。このように，媒体上にデータを記録することを書き込みといい，また媒体上のデータを再生することを読み出しという。

磁気ヘッドは，**第2.25図**のように，巻線が巻かれているパーマロイなどの強磁性材料の鉄心の一部に，きわめてせまいギャップを設けたものである。このギャップを，**磁気ギャップ**と呼んでいる。

いま，図(**a**)のように，磁気ヘッドの巻線に電流を流したとき生ずる磁束は，磁気ギャップの個所で外部へ漏れ，その近くに漏れ磁界が発生する。また，図(**b**)のように逆方向の電流を流すと，逆方向の漏れ磁界が発生する。このように，巻線に流す電流の方向によって磁界の方向が制御される。

前述の磁気記録媒体を磁気ヘッドの近くに置けば，この漏れ磁界によって媒体の

(1)　magnetic head



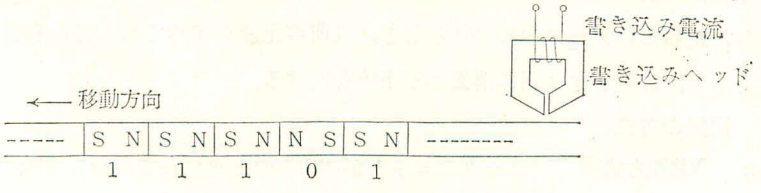
■■■は，磁気記録媒体の磁化状態（電流の方向と媒体上の磁化方向に注意のこと）

第2.25図　磁気ヘッドによる書き込み

一部が磁化され，しかも磁気ヘッドに流す電流の方向によって媒体上の磁化方向が制御されることになる。

この媒体上の2つの磁化方向を，それぞれ“1”および“0”に対応させれば，1つの磁化点で1ビットが記録されると考えることができる。

ここで，**第2.26図**のように記録媒体上を磁気ヘッドを移動させるなどして，両者の相対位置を変えながら，磁気ヘッドに流す電流の方向を，書き込みたいデータによって変化させれば，データに対応した多くのビットを媒体上に記録していくことができる。この磁化された状態は，きわめて小さな数多くの永久磁石が，あるものはＮＳまたあるものはＳＮというように，媒体上に並んでいると考えればよい。



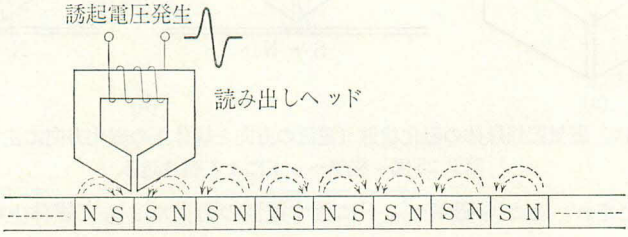
ＳＮを“1”，ＮＳを“0”としてある。

第2.26図　記録媒体上の記録

以上が書き込みの原理で，このとき使用する磁気ヘッドを**書き込みヘッド**(1)といい，書き込みヘッドの巻線に流す電流が書き込み電流である。

(1)　writing head

つぎに，読み出しであるが，媒体表面に書き込まれたビットは，磁化されたときの状態を持続しているので，それを磁気ヘッドによって検出する。磁化の方向が場所によって異なっている磁気記録媒体を磁気ヘッドに近接して一定速度で通過させると，媒体から出て磁気ギャップを横切る磁束が時々刻々と変化するので，磁気ヘッド内の磁束の変化に比例した誘起電圧が巻線に発生する（**第2.27図**参照)。



第2.27図　読み出し原理図

このように，読み出しに使用する磁気ヘッドを**読み出しヘッド**(1)といい，巻線に発生した誘起電圧によって生ずる電流が読み出し電流である。

磁気記録媒体に記録されたデータを読み出す場合は，磁心に記憶されているデータの読み出しの場合とは異なって，読み出しによって記録内容が破壊されることがない。このような読み出しが，**非破壊読み出し**である。

媒体上の記録は，何回読み出してもその内容は不変であるが，新らしいデータを書き込めば，前に記録されていたデータが消えて新らしい内容に書き替えられる。

なお，新らしいデータを書き込むとき，以前の記録をすべて消してから書き込む場合もあり，このときは別に**消去ヘッド**(2)を設ける。

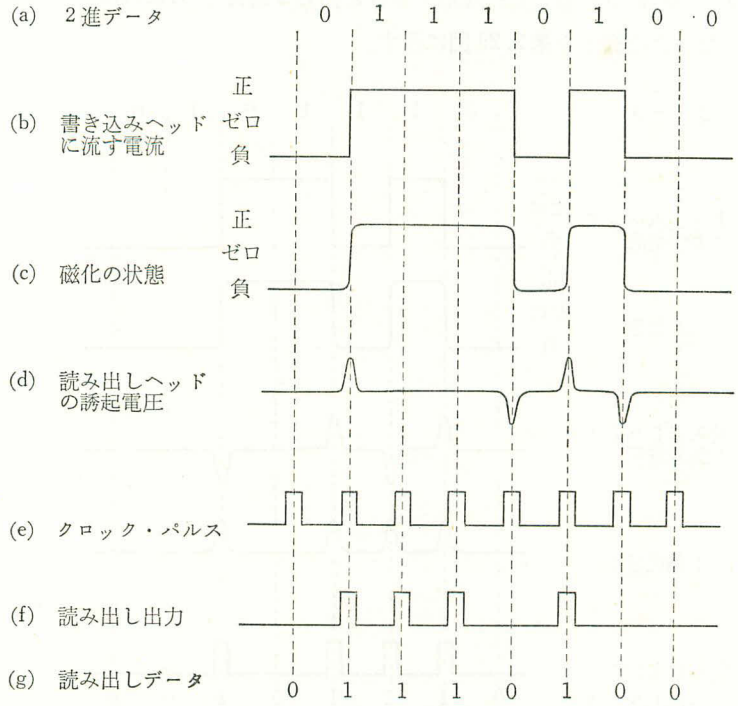
2.　記録の方式

a.　NRZ方式(3)　　データのビットが“1”のときは正方向に，また“0”のときは負方向に書き込み電流を流して記録する方式である。“1”あるいは“0”が連続しているときは，書き込み電流も連続して流しゼロになることはない。したがって，磁気記録媒体上の磁化の状態も，正負のいずれかであって，ゼロになること

(1)　reading head　　(2)　erasing head
(3)　non return to zero method：略して NRZ

はない。これが **NRZ 方式**といわれる理由である。

一例として，2進符号 01110100 を NRZ 方式により記録するときの方法を**第 2.28 図**に示す。

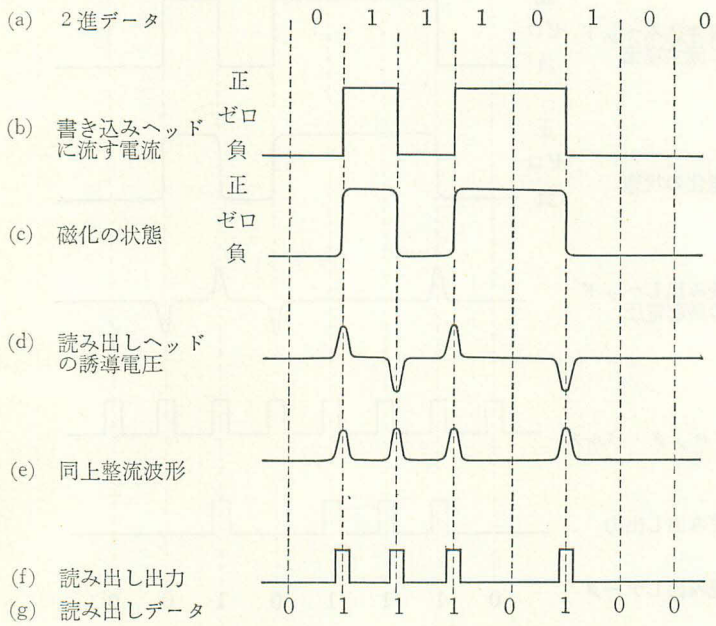


第 2.28 図 NRZ方式

NRZ 方式で記録されているデータを読み出すとき，磁化状態が正→負，あるいは負→正に変化している個所で，読み出しヘッドに誘起電圧が発生する。そこで，**クロック・パルス**(1)（一定時間ごとに発生するパルスで，**1.5.2** で述べた同期信号がこれに相当する）を利用して読み出し出力を得る。すなわち，読み出しヘッドの誘起電圧が正になった瞬間から，負になる瞬間のすぐ前までのクロック・パルスを“1”の出力として取り出すことによりデータが復元される。

(1) clock pulse

b. NRZI 方式　この方式も NRZ 方式の一種であるが，“1”を書き込むときだけ書き込み電流の方向を反転し，“0”のときは電流の方向を変化しない方式である。したがって，磁化の状態が変化する個所には“1”が，また，変化のない点には“0”が記録されることになる。さきと同じ2進符号 01110100 を NRZI 方式で記録したときの様子を**第 2.29 図**に示す。

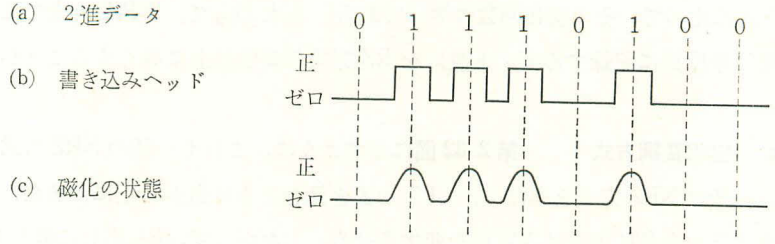


第 2.29 図　NRZ 方式（NRZI 方式）

NRZI 方式で記録されているデータを読み出すときは，磁化の状態が変化している個所，すなわち，“1”が記録されている個所で，読み出しヘッドに電圧が誘起する。これを全波整流して，図の(f)の読み出し出力を得ることができ，データが復元される。

c. 2レベル RZ 方式[1]　**第 2.30 図**のように，正およびゼロの2レベルを使用する方式で，データのビットが“1”のときだけ正方向に書き込み電流を流し，

(1) 2-level return to zero method

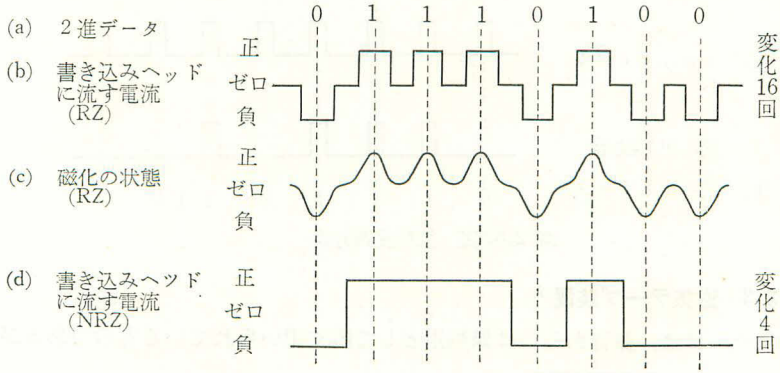


第2.30図 2レベルRZ方式

“0”のときおよびビットとビットの間は電流は流さない(ゼロのまま)方式である。したがって，記録媒体も“1”のときだけ磁化され，その他のときは磁化されない。このように，“1”以外のとき磁化がゼロになるので，RZ方式といわれている。

d. 3レベルRZ方式 この方式は，第2.31図に示すように，正，負およびゼロの3レベルを使用するRZ方式で，データのビットが“1”のとき正方向に，また“0”のとき負方向に書き込み電流を流し，ビット間は電流を流さない（ゼロのまま）方式である。したがって，媒体上では，“1”のビットは正方向に，“0”のビットは負方向に磁化され，ビット間はどちらの方向にも磁化されない。

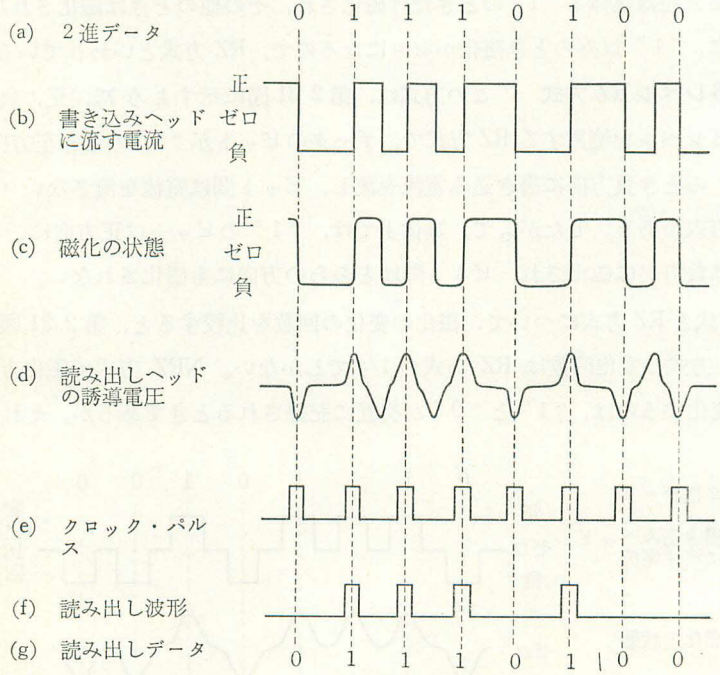
NRZ方式とRZ方式について，磁化の変化の回数を比較すると，第2.31図の場合，NRZ方式の変化回数はRZ方式の1/4でしかない。NRZ方式で磁化方向が最も多く変化するのは，“1”と“0”が交互に記録されるときであるが，それでも



第2.31図 3レベルRZ方式とNRZ方式

RZ 方式に比べて，その変化回数は半分である。したがって，NRZ 方式では，記録密度（単位長に記録するビット数）を RZ 方式の 2 倍以上に高くすることができる。

e. 位相変調方式(1)　第 2.32 図に示すように，これも一種の NRZ 方式であるが，一般の NRZ 方式と異なり，“1”を書き込むときは負から正に，また，“0”のときは正から負に書き込み電流を変化させる。したがって，読み出しに際しては，ビットが連続していても必ず誘起電圧を発生し，その状態は交流波形に近くなるので，交流回路が使用できるのが特徴である。



第 2.32 図　位相変調方式

2.3.3 磁気テープ装置(2)

磁気テープは，録音あるいは録画用として広く用いられているものであるが，電

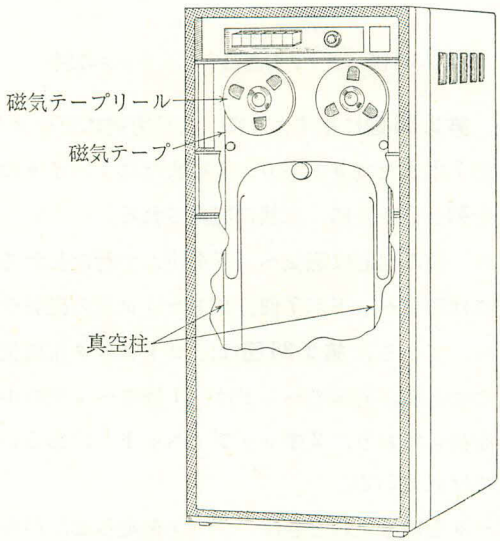
(1) phase modulation method　(2) magnetic tape unit

子計算機でもデータの記録媒体として広く使用されており，電子計算機の漫画では磁気テープ装置が必ず出てくるほどである。

磁気テープ1巻には，約700 mの磁気テープが巻かれており，1巻当たり約1000万字以上のデータが記録できる。さらに，磁気テープは，磁気テープ装置に自由に着脱できるので，何巻かの磁気テープを使うことによって，きわめて大量のデータを記録することができる。また，磁気テープ装置の1ビット当たりの価格はきわめて安い。こうしたことが，磁気テープ装置が電子計算機の補助記憶装置として広く使用されている理用である。

しかし，磁気テープでは，ほしいデータを読み出すまでにかなりの時間（たとえば，テープの終わりの方のデータを読み出すには数分）を必要とするのが欠点である。なお，磁気テープ装置のように，テープの始めから順にデータの読み書きをする装置を，**シーケンシャル・アクセス装置**(1)といっている。

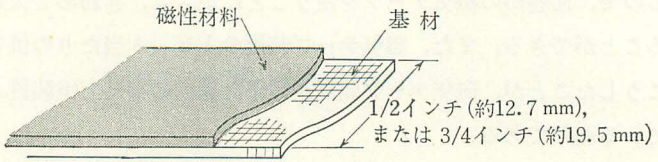
第2.33図に，磁気テープ装置の外観を示す。



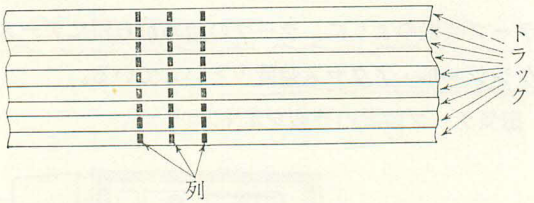
第2.33図　磁気テープ装置外観

(1) sequencial access unit

1. データの書き込みと読み出し　磁気テープは，**第2.34図**に示すように，厚さ38μ，幅1/2 in（12.7 mm）または3/4 in（19.5 mm）のポリエステルなどを基材とし，その上に磁性材料として酸化鉄の微細粉子を薄く(10μ)塗布したものである。この磁気テープ上に，データが記録される様子はつぎのとおりである。



第2.34図　磁気テープ



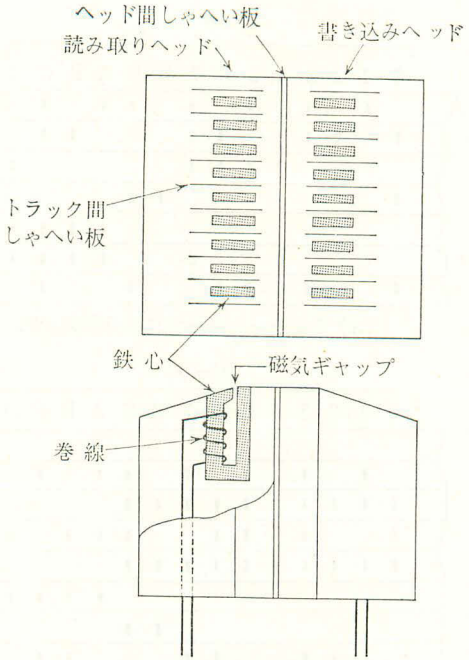
第2.35図　データの記録（9トラックの例）

磁気テープには，**第2.35図**に示すように，長尺方向にデータを記録する道（**トラック**(1)という）が7本または9本あり，1字または1バイトのデータは磁気テープの幅方向（これを**列**という）に，点状に記録される。

データの書き込み，読み出しは磁気ヘッドを使って行なわれるが，7トラック用の磁気テープ装置には磁気ヘッドが7個，9トラック用の磁気テープ装置には磁気ヘッドが9個設けられている。**第2.36図**は，9トラック用磁気テープ装置の磁気ヘッドを示したものである。同図のヘッドは，1個のヘッドの中に書き込みヘッドと読み出しヘッドを持っており，**2ギャップ・ヘッド**(2)，あるいは**デュアル・ギャップ・ヘッド**(3)と呼ばれている。

磁気テープにデータを書き込むときは，テープを走らせながら7あるいは9個の書き込みヘッドの巻線に書き込み電流を流し，磁気テープ上の7あるいは9トラッ

（1） track　（2） 2-gap head　（3） dual gap head



第2.36図　9トラック用磁気ヘッド

クを順次点状に磁化して行く。

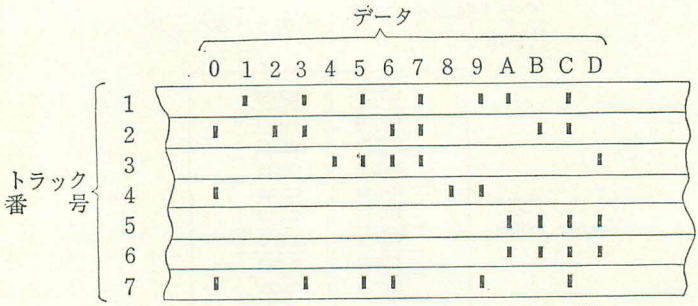
同様に，磁気テープからのデータの読み出しは，読み出しヘッドの巻線に発生する誘起電圧によって生ずる読み出し電流を検出して行なう。

書き込まれたデータは，テープの走行に伴って読み出しヘッドの個所にきたとき読み出され，パリティ・チェックにより，正しく書き込まれたかどうかの検査が行なわれる。

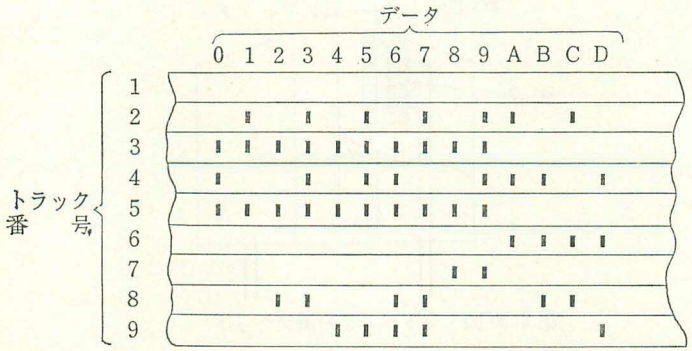
磁気ヘッドには，1つのギャップで読み出しと書き込みの両方行なうことのできる**シングル・ギャップ・ヘッド**(1)もあるが，ギャップが1つであるため，書き込みと読み出しを同時に行なうことはできない。そのため，書き込まれたデータを検査するには磁気テープを少しもどして，改めて読み出して検査する必要がある。

第2.37図 (**a**) は，7トラックの磁気テープに，また，同図 (**b**) は9トラックの

(1) single gap head



(a) 7トラックテープによる記録一例



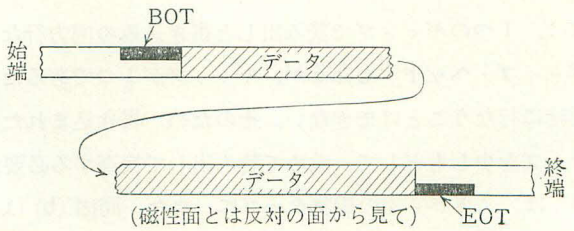
(b) 9トラックテープによる記録の一例

▮は“1”に磁化されていることを示す。

第2.37図　データの記録

磁気テープに，英字と数字が記録されている様子を示したものである。

2.　磁気テープ上のデータの記録　　第2.38図に示すように，磁気テープには，

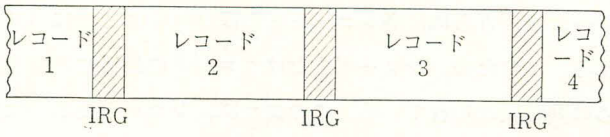


第2.38図　BOT と EOT

その始端から数 m の位置の磁性面とは反対の面にアルミはくがつけられている。これを **BOT**(1) という。また，終端から数 m の位置にも，**EOT**(2) と呼ばれるアルミはくがつけられている。データは，この BOT と EOT の間に記録される。

磁気テープへのデータの書き込みや読み出しは，**レコード**(3)単位あるいはいくつかのレコードを集めた**ブロック**(4)の単位で行なわれる（ここで，レコードとは，一連のデータをある大きさにまとめたものである）。

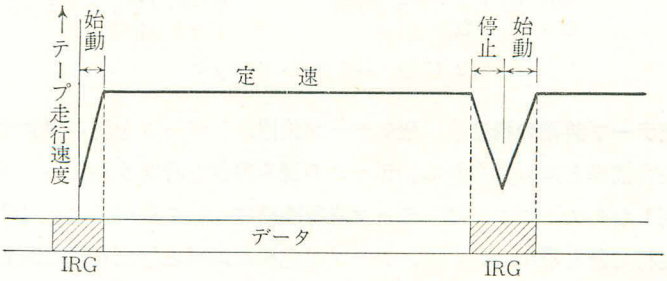
1つのレコードとつぎのレコードの間，あるいは1つのブロックとつぎのブロックの間には，**第 2.39 図**に示すように **IRG**(5)，あるいは **IBG**(6) と呼ばれる空白（何も記録されない）の部分が設けられている。この IRG あるいは IBG は，データの



（a）IRG～IRG 間にレコード1個

（b）IRG～IRG 間に複数個レコード

第 2.39 図　IRG の様子

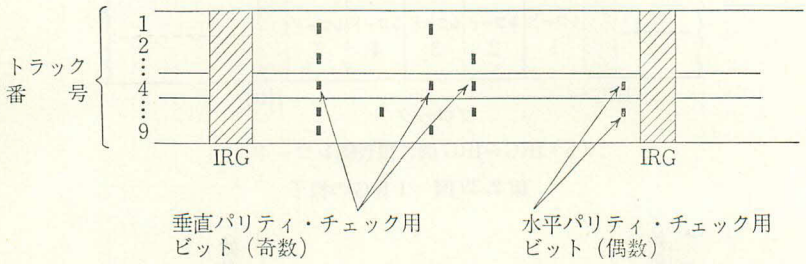


第 2.40 図　テープの進行

(1) beginning of tape の略　(2) end of tape の略　(3) record
(4) block　(5) inter-record gap の略　(6) inter-block gap の略

区切りをはっきりつけるためと，データを読むときにテープが規定の走行速度に達するのに必要な助走距離と，読み終わって停止するまでに必要な距離を確保するために設けられるもので，書き込み動作の際作られ，普通約 2cm 程度である。**第 2.40 図**は，磁気テープ走行速度と IRG の関係を示したもので，IRG で止まっている磁気テープが始動し，規定速度に達したのちデータの読み出しが行なわれ，つぎの IRG で停止するまでの様子が示されている。

磁気テープ上に記録されたデータは，1字ずつパリティ・チェックが行なわれる。**第 2.41 図**の4番のトラックが，パリティ・ビットを記録するためのトラックである。このように，1字ずつ行なう検査を**垂直パリティ・チエック**という。磁気テープ上のデータは，さらに，各レコードまたはブロックごとにパリティ・チェックが行なわれる。このため，レコードまたはブロックの最後にパリティ・ビットを記録するための列が設けられている。このように，レコードあるいはブロックごとに行なう検査を**水平パリティ・チェック**と呼んでいる。



第 2.41 図　パリティ・チェック

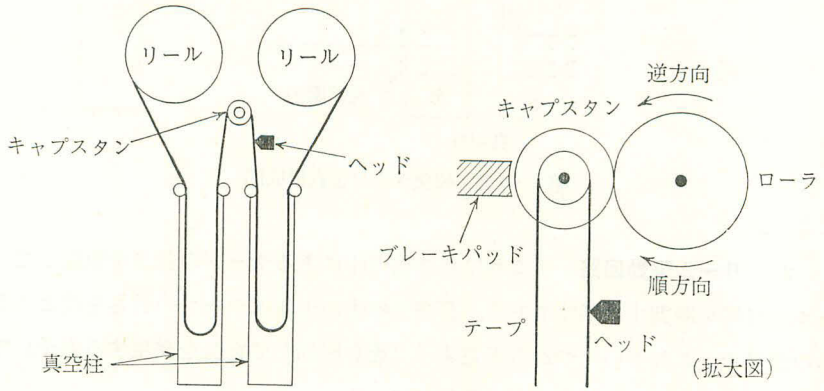
3.　磁気テープ装置の構造　磁気テープ装置は，データを読み書きする機構とテープの走行機構とに大別される。データの読み書きを行なう磁気ヘッドについては先に説明したので，ここでは，テープ走行機構について述べることにする。

テープ走行で最も難かしいのは，テープが急激な走行と停止を繰り返しながら，読み書きの動作を進めていく点である。IRG の中央に止まっている磁気テープが走行を開始する場合，約 1 cm 進んだときには 2～3 m/s の一定速度になっている必要があり，また，走行中の磁気テープが停止する場合も急激に停止する必要があ

る。家庭用テープ・レコーダ（テープ走行速度は，4.75～19 cm/s）のようなテープ走行機構で，このような急激な走行，停止を行なわせようとすれば，テープ・リールの慣性によってたちまちテープ切断などの事故が生ずるであろう。

そこで磁気テープ装置では，走行機構に種々の工夫がなされている。

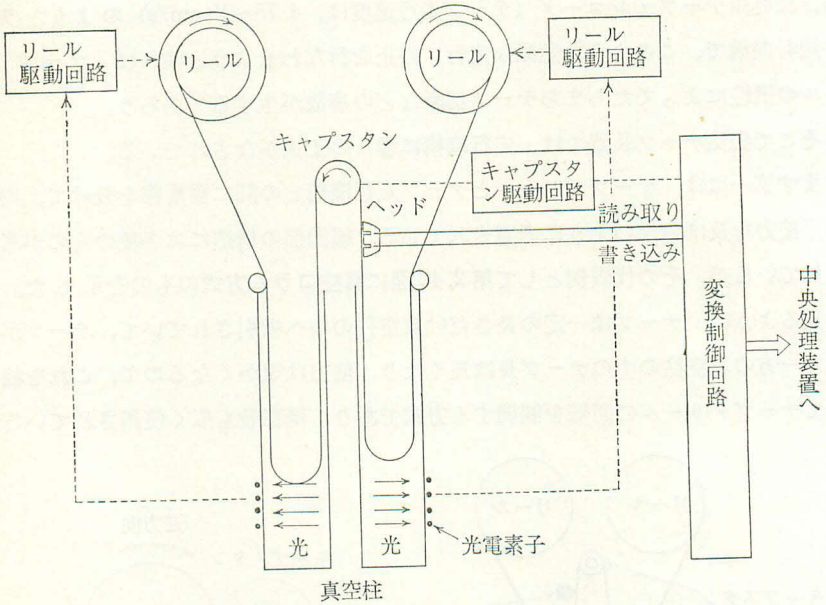
まず第一には，テープ・リールとテープ走行機構との間に緩衝部を設けて，両者が直接力を及ぼさないように考慮されている。緩衝部の構造には各種のものが考えられているが，その代表例として**第2.42図に真空コラム方式**のものを示した。図に見るように，テープは一定の長さだけ真空柱の中へ吸引されていて，テープが走ると一方の真空柱の中のテープ長は長くなり，他方は短かくなるので，これを検出してテープ・リールの回転を制御する方式であり，現在最も広く使用されている。



第2.42図　真空コラム式緩衝機構とシングル・キャプスタン方式

第二の工夫は，直接テープの走行をつかさどるキャプスタンである。ここでは各種方式のうち最も簡単な**シングル・キャプスタン方式**について説明すると，第2.42図に示すように，1個のキャプスタンとローラがあり，テープを順方向に走らせるときはローラを順方向に，また，逆方向に走らせるときは逆方向に回転させ，これにキャプスタンを押しつけることによりテープの走行が行なわれる。テープには真空柱により一定の張力が与えられているので，キャプスタンが回転すれば，その間の摩擦によってテープが走り出す。

磁気テープ装置の構成図を**第2.43図**に示す。各部の機能の概略はつぎのごとく



第2.43図　磁気テープ装置構成図

である。

a.　リール駆動回路　これは真空柱の中にあるテープの長さを監視して，リールの回転を制御する回路である。真空柱の中の矢印はテープの長さを検出するための光線で，これをいくつかの光電素子（光を感知して電流を発生する素子）で受けることにより，テープの長さを監視している。

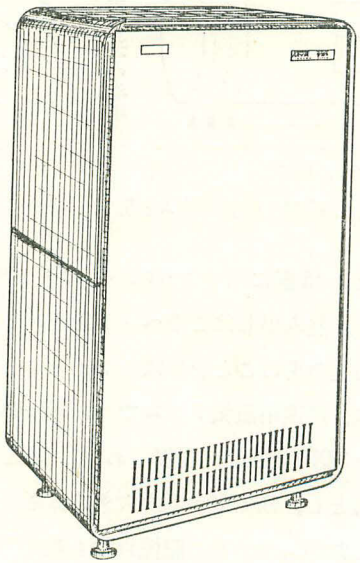
b.　キャプスタン駆動回路　これはキャプスタンの回転を制御して，テープを走行させるための回路である。図はテープが順方向に走行している状態で，逆方向に走行させるときは，キャプスタンは逆回転する。

c.　変換制御回路　これには磁気テープから読み出したデータを中央処理装置へ送ったり，あるいは中央処理装置から送られてきたデータを受けてヘッドへ電流を流す回路などが組み込まれている。

2.3.4　磁気ドラム記憶装置(1)

(1)　magnetic drum memory

磁気ドラム記憶装置は，初期の電子計算機の主記憶装置に用いられたこともあるが，磁心記憶装置が主記憶装置の座を占めるにつれて，補助記録装置として用いられるようになった。その外観を**第2.44図**に示す。

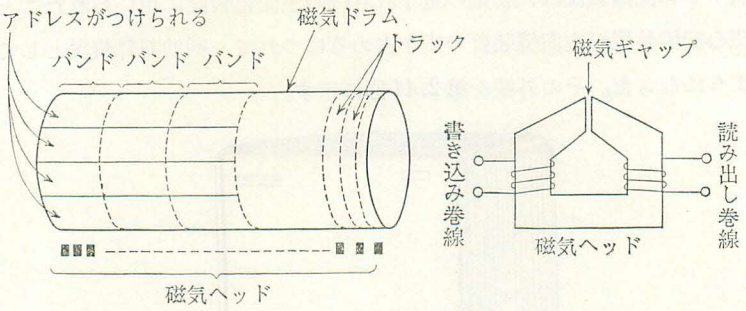


第2.44図　磁気ドラム記憶装置

磁気ドラム記憶装置は，磁性材料を塗布またはめっきした金属円筒を高速で回転させ，その周辺に設けられた磁気ヘッドを使って，円筒表面にデータを書き込んだり読み出したりする装置である。

磁気テープ装置が始端から順にデータの読み書きをする方式（シーケンシャル・アクセス方式）であったのに対し，磁気ドラム記憶装置では，データを書き込むべき場所，あるいは読み出したいデータの記録されている場所を，その順番に関係なく，すばやく見つけることができるのが特徴である。このような装置を，**ランダム・アクセス記憶装置**あるいは等速呼び出し記憶装置という。

1. 磁気ドラム　磁気ドラムは直径 10～30cm，高さ20～40cm 程度のアルミ合金製の円筒表面に酸化鉄の微細粒子を塗りつけたり，ニッケル・コバルト合金をめっきしたものである。



第2.45図　磁気ドラムと磁気ヘッド

このドラムに沿って，第2.45図に示すようにいくつかの磁気ヘッドが配置されていて，データの書き込み，読み出しはこのヘッドを使って行なわれる。データを書いたり読んだりする円周上の帯のことをやはりトラックという。

磁気ドラムの回転速度は，高速用磁気ドラムで 10 000～20 000 回転/分程度，低速用磁気ドラムで 1 500～3 000 回転/分程度である。高速用磁気ドラム記憶装置は，高速の読み書きを目的とし，磁気ドラムの大きさは直径 10 cm 程度，高さ 10～20 cm 程度と小さく，したがって，その記憶容量も数十万字程度と小さい。これに対して低速用磁気ドラム記憶装置は，読み書きの速度を上げることよりも，記憶容量を大きくすることを目的とし（記憶容量は数百万字程度），したがって，ドラムも大きくなっている。

磁気ドラムへのデータの書き込みは，ヘッドの書き込み巻線に書き込み電流を流すことにより生じた磁界がヘッド先端のギャップから漏れ，ドラムのトラックを磁化することによって行なわれる。

また，データの読み出しは，トラック上の磁界がヘッドに磁束を発生させ，読み出し巻線に読み出し電流を誘起することによって行なわれる。

ランダム・アクセス記憶装置は，データの記録されている場所に関係なく，すばやくデータを見つけだせるのが特徴であるが，厳密には記憶場所により，ヘッドの位置に到着するまでの時間に若干の差異がある。

ドラムが 3 000 回転/分で回転しているとすると，1回転に要する時間 T は，

$$T=\frac{60\times1000}{3000}=20\ \text{ms} \tag{2.1}$$

となる。したがって，あるトラック上のデータを読み出すとき，それがヘッドのすぐ下にあるときはすぐ読み出すことが可能であるが，ヘッドを通過した直後の場合にはドラムが1回転するまで20 msの間待たなければならない。こうして，データの読み出しには，平均して10 ms必要ということになる。この待ち時間のことを，アクセス・タイムあるいは呼び出し時間という。

2. **磁気ドラム上のデータの記録**　磁気ドラム上のデータの記録形式には，大別して，つぎの3種類がある。

（1）　**ビット直列形式**　この形式は，**第2.46図 (a)** のように，1つのトラックにビットを直列に並べて記録するものである。

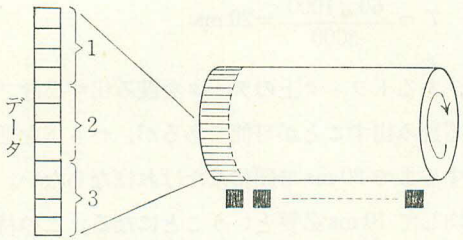
（2）　**ビット並列形式**　この形式は，同図 (b) のように，軸方向の何トラックかを使って，ビットを並列に並べて記録するものである。

（3）　**ビット直並列形式**　これは，上記2者の混合で，2進化10進数（BCD）の記録などに有効な形式である。2進化10進数1けたを表わすのに4ビットなので，同図 (c) のように，軸方向の4トラック（これを**バンド**(1)という）を使って1けたを記録し，これを円周方向に直列に並べることにより，順次1けたずつ記録していく(2)。たとえば，2進化10進数の10けたで構成されているデータは，1バンド4トラック上に10けた直列に並べて記録すればよい。

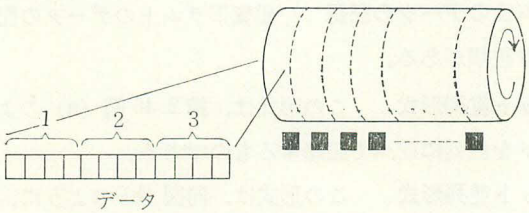
これらの3種類の形式のうち，どれを選ぶかには，データを読み書きするに要する時間とヘッドの数が関係する。たとえば，2進化10進数で表わされた123を書き込むとき，第2.46図に示すように，直列形式ではヘッドは1個でよいが，書き込みには12ビットぶんの時間を必要とするのに対し，並列形式では書き込みは1ビットぶんの時間で行なうことができるが，ヘッドは12個必要とする。一方，直並列形式では，ヘッドは4個であり，書き込みも3ビットぶんの時間ですむ。すなわち，

（1）　band

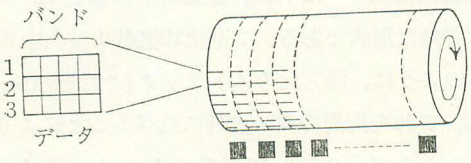
（2）　実際には，チェック用のトラックも必要なので，2進化10進数を記録するには1バンドは5トラックになる



（a）　ビット直列



（b）　ビット並列



（c）　ビット直並列

第2.46図　記　録　形　式

ヘッド数は並列形式の 1/3，また，書き込みに要する時間は直列形式の 1/4 ですむことになる。現在のドラムでは，この直並列形式が多く使われている。

直並列形式の場合，ドラム表面は第 2.45 図のように，軸方向はバンドによって，また円周方向は 1 語（たとえば 2 進化 10 進数の 10 けたというような）ごとに区切られており，表面は多くの小区画に分けられているとみなすことができる。この区画のそれぞれにつけられた番号が，アドレス（あるいは番地）であり，データの書き込み，読み出しは，このアドレスを指定して行なわれる。

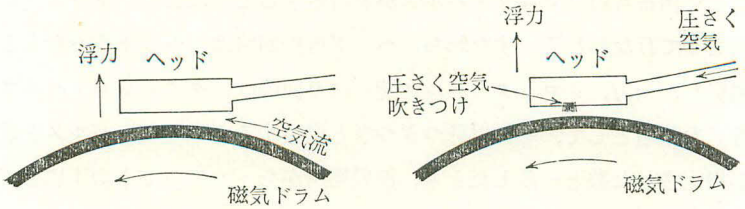
3.　磁気ドラム記憶装置の構成　　第 2.47 図のように，ドラムとヘッドとの間

には，ドラム表面がヘッドによって傷つけられないように，わずかな空隙が設けられている。ドラムが回転すると，ドラム表面の空気が移動し，それに伴う浮力によって，ヘッドは常に一定間隙をもって，ドラム表面から浮いた状態になっている。のような方式のヘッドを**浮動ヘッド**(1)という。浮動ヘッドには，このほか，ヘッドから圧さく空気をドラム表面に吹きつけて浮力を得る方式のものもある。

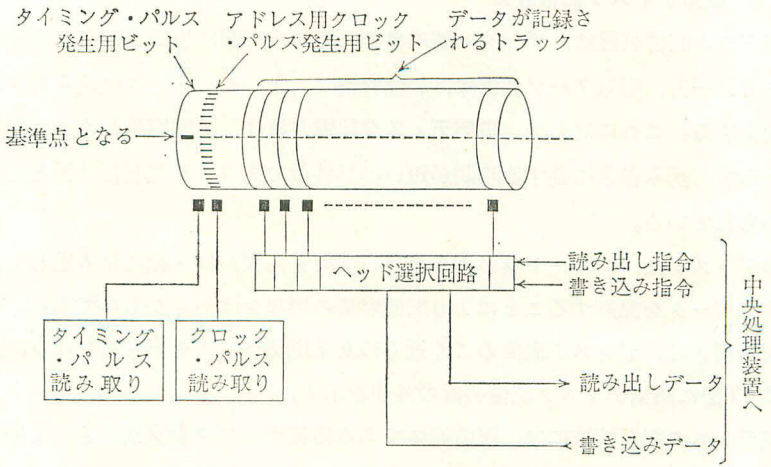
第2.48図は，磁気ドラム記憶装置の構成図である。

磁気ドラムには，データを記録するトラックのほかに，2つのトラックが設けられている。

1つはタイミング・パルスを発生するためのトラックで，この上に1ビットだけ



第2.47図 浮動ヘッドの例



第2.48図 磁気ドラム記憶装置構成図

(2) floating head

が記録されている。ドラムが1回転するたびに1回このビットが読み出され，タイミング・パルスを発生する。このパルスが発生した時点を，書き込み，読み出しの基準とする。

もう1つのトラックには，アドレス用のクロック・パルスを発生するビットが記録されている。これは，たとえば円周方向に100語記録する方式のドラムであれば，100個のビットが等間隔で記録されている。したがって，タイミング・パルスを検出した時点を基準としてクロック・パルスを数えれば，いま何語目のデータが記録されている場所であるかを知ることができる。

磁気ドラムに対するデータの読み書きは，ヘッドの番号と，タイミング・パルスを基準にして何番目のクロック・パルスかを指定することにより，ドラム上のアドレスを指定して行なわれる。すなわち，ヘッド選択回路は，ヘッド番号をもとにヘッドを選択し，一方，クロック・パルス読み取り回路は，タイミング・パルスが発生した時点を基準として，それ以降つぎつぎと発生するクロック・パルスを数え，その数が指定された数と一致したとき，記録場所がちょうどヘッドの下にきていることになり，さきに選択したヘッドによって，読み書きが行なわれる。

2.3.5　磁気ディスク記憶装置(1)

磁気ドラム記憶装置は，データの読み書きに要する時間は短いが，記憶容量に限界があり，一方，磁気テープ装置は，記憶容量は大きいが，データの読み書きに長時間を要する。これに対して，**磁気ディスク記憶装置**は，記憶容量が大きいのにもかかわらず，読み書きに要する時間が短いのが特徴であり，補助記憶装置として広く用いられている。

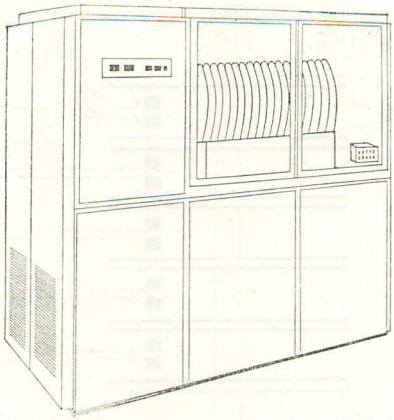
磁気ディスクは，レコード盤のような薄い円板を何枚か同一軸に積み重ねて，その表面にデータを記録することにより記憶容量の増加をはかったものである。データの読み書きは，ディスク表面のごく近くにある磁気ヘッドを使って行なう。

第2.49図に磁気ディスク記憶装置の外観を示す。

磁気ディスク記憶装置には，記憶媒体である磁気ディスクを交換できるものと固定のものとがあり，前者を特に**ディスク・パック**(2)と呼んでいる。

---

(1)　magnetic disk memory　　(2)　disk pack



第2.49図 磁気ディスク装置

ディスク・パックは，直径約35 cmの円板を普通6枚重ねてこれを回転軸に装着し，円板の表面にデータを記憶するものである。ディスク・パックは容易に着脱できるので，いくつかのディスク・パックを使うことにより大量のデータを記録することができる。

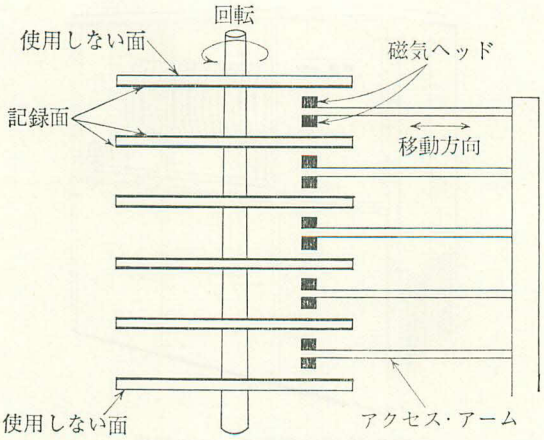
ディスクが固定されているものは，直径1 m程度の大きな円板を20～数十枚重ねて，回転する水平軸に取りつけ，円板の表面にデータを記録する。

ディスクの基板としては，アルミ合金などの円板が用いられ，また，磁性材料としては基板表面に酸化鉄の微細粒子を塗布したり，ニッケル・コバルトをめっきしたものが用いられる。

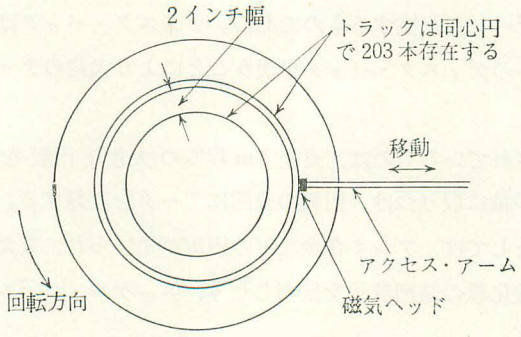
1. ディスク・パック ディスク・パックは第2.50図のように，通常6枚の円板（ディスク）が1つの軸に取りつけられていて，これを装置の回転軸に装着する。デイスク面は12面あるが，最上面と最下面はほこりやきずがつきやすいので，この2面を除く10面にデータが記録される。ディスクの1記録面に1個の磁気ヘッドが用意され，これが**アクセス・アーム**(2)といわれる腕に取りつけられている。

ディスク面にデータを読み書きするときは，アクセス・アームが動いて，磁気ヘッドを所定の位置まで移動させる。

(2) access arm



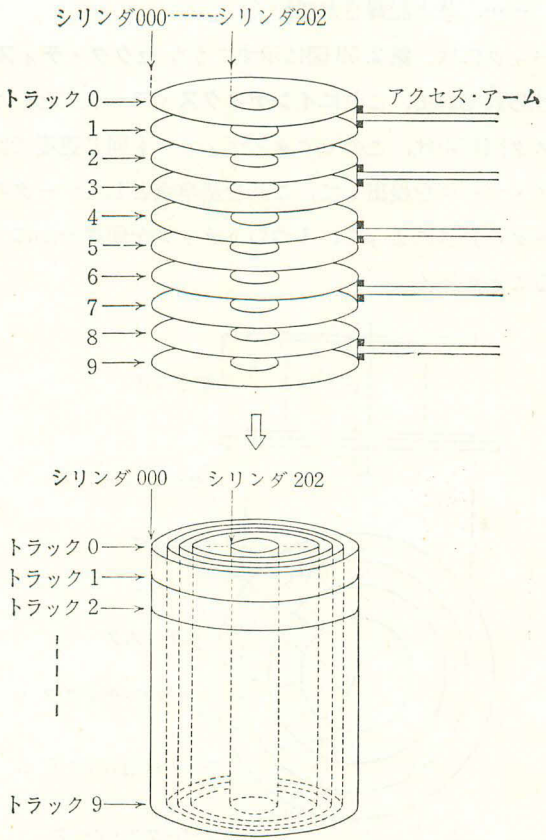
第2.50図　ディスクとアーム



第2.51図　トラック

いま，1ディスク面を考えると，**第2.51図**のように，磁気ヘッドがディスク上のある位置で停止していてディスクが1回転すると，面上に1本の帯状の輪ができる。これもトラックといわれ，データはこのトラック上に記録される。ディスク・パックでは，1枚のディスク表面上2インチの幅の中に，このような同心円状のトラックが，通常203本存在する。

つぎに，10ディスク面について考えてみる。10個の磁気ヘッドがある位置で止まっていて，6枚のディスク（10面）が1回転すると，**第2.52図**のように，10個の磁気ヘッドによって作られるトラックが縦方向に10本でき，あたかも10本の円



第 2.52 図　シリンダとトラックの関係

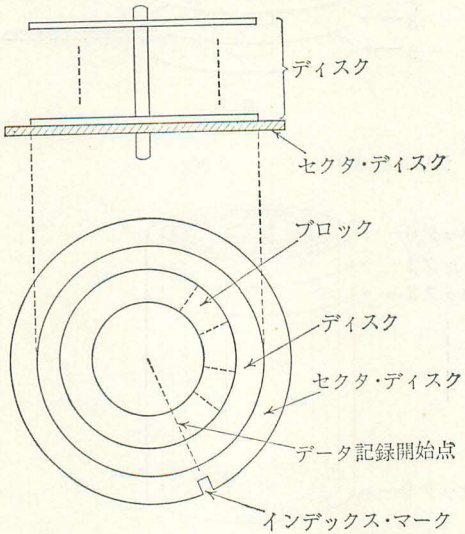
周方向の筋をもった円筒を想定することができる。これを**シリンダ**といい，仮想円筒上の 10 本の筋をトラックという。したがって，ディスクの記録面全体では，このようなシリンダを 203 個考えることができる。

第 2.52 図において，トラックは，ディスク面の上からトラック番号 0，1，2，……，9 の番号がつけられ，シリンダは，外側から内側へシリンダ番号 0，1，2，……，202 と番号づけされる。

データは，普通 1 つのシリンダのトラック番号 0 から順次 1，2，……，9 と記録され，1 シリンダ分の記録が終わるとつぎのシリンダに移って，前と同様トラッ

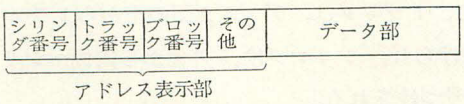
ク番号0，1，……，9と記録されていく。

ディスク・パックには，第2.53図に示すような**セクタ・ディスク**(1)といわれる別の円板が設けられていて，これに**インデックス・マーク**(2)という刻み目がはいっている。デイスク回転中は，このセクタ・ディスクも同じ速度で回転するので，このインデックス・マークを検出して，これを基準点としてデータの読み書きを行なう。また，データの長さによって，1つのトラックを何等分かに分割したブロックの中に記録することがある。



第2.53図　セクタ・ディスクとインデックス・マークおよびブロック

記録面上での記録方式には各種あるが，普通第2.54図に示すように，アドレス表示部と，データ部とからなっている。アドレス表示部には，データの記録位置を示す情報，すなわち，シリンダ番号，トラック番号，ブロック番号などが記録され



第2.54図　データの記録形式

(1)　sector disk　　(2)　index mark

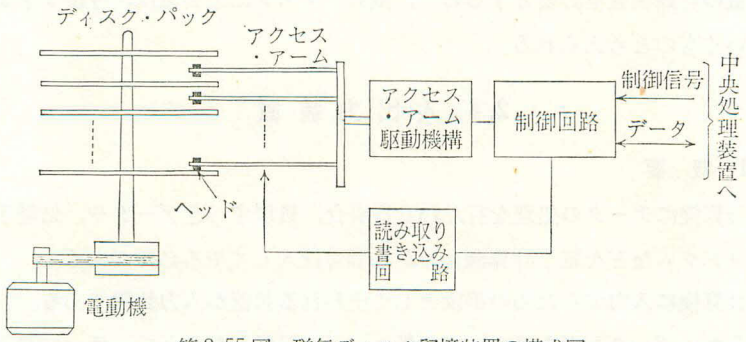
ている。したがって，プログラムでシリンダ番号，トラック番号およびブロック番号を指定すれば，データの記録位置を指定することができる。この場合，まずヘッドを指定のシリンダの位置まで移動させ，ついで回転中のディスクの所定のブロックがヘッド位置にきたとき，トラック番号で指定されている磁気ヘッドを用いて読み書きすることができる。

このことからも明らかなように，磁気ディスクのアクセス・タイムは，ヘッドを所定の位置まで移動するに要する時間と，所定の記録位置がヘッドの下にくるまでの時間の和となる。したがって，磁気ドラム記憶装置のアクセス・タイムが 10 ms 程度であるのに対し，磁気ディスク記憶装置のアクセス・タイムは 100 ms 程度になっている。

しかし，磁気テープ装置の数分に比べれば，はるかに速く読み書きできる。

また，1ディスク・パック当たりの記憶容量は，数百万～千万字程度で，磁気ドラム記憶装置より大きく，磁気テープ（千万字）とほぼ同程度の記憶容量を有している。

2. 磁気ディスク・パック記憶装置の動作　第 2.55 図に，磁気ディスク・パック記憶装置の構成図を示す。



第 2.55 図　磁気ディスク記憶装置の構成図

磁気ヘッドは，磁気ドラム装置の項で説明したような浮動ヘッドが多く用いられており，ディクス面を傷つけないよう考慮されている。

アクセス・アーム駆動機構は，磁気ヘッドを所定の位置まで迅速に移動させるた

め，磁気ヘッドが取りつけられているアクセス・アームの運動を制御する機構で，油圧シリンダの油の量を制御して，その油圧によってアクセス・アームを動かす方式のものや，モータの回転によってアクセス・アームを動かす方式などがある。

磁気ディスクに対するデータの読み書きは，さきに触れたように，シリンダ番号，トラック番号およびブロック番号によって指定される。

まず，中央処理装置から「アーム移動」の指示がくると，ヘッド位置決め機構が働いて，シリンダ番号で指定されている位置までアクセス・アームを動かす。ついで，中央処理装置から「読め」あるいは「書け」という命令がくると，トラック番号によって指定されている磁気ヘッドを選択し，ブロック番号では指定されている記録個所がヘッドの下にきたとき，その磁気ヘッドを用いて読み書きを行なう。

磁気ディスク記憶装置の大容量化は，現在も積極的に進められており，たとえば円板の数を増やしたディスク・パックを数台組み合わせたうえ，記録密度を従来のものより大きくすることにより2～数億字を記録することができる**大容量ディスク・パック装置**が開発されている。

電子計算機と通信回線とを結んで，即時にデータ処理を行なう，いわゆる**オンライン・リアルタイム**(1)処理（第3章参照）では，データの迅速な読み書きを行なえる大容量の記録装置を必要とするので，磁気ディスク記憶装置は，今後ますます進歩していくものと考えられる。

2.4 入出力装置

2.4.1 概　要

電子計算機にデータの処理を行なわせる場合，処理すべきデータや，処理手順を示すプログラムなどを電子計算機がわかる形で投入してやる必要がある。

電子計算機に入力するための手段として使われる装置が入力装置である。

入力したいデータなどを紙テープや紙カードに記録したうえで，テープ読み取り機やカード読み取り機にかけると，データは電気パルスに変換されて中央処理装置へ送られ，記憶装置で記録される。紙テープ読み取り機や紙カード読み取り機は，

(2)　on-line real time

現在用いられている入力装置の代表的なものである。

一方，電子計算機で処理された結果は，人間にわかる形，たとえば印刷物として外部に表わさなければならない。

電子計算機から出力するための手段として使われる装置が出力装置である。

記憶装置に記憶されている出力すべきデータは，電気パルスに変換されて出力装置へ送られ出力される。行印字機は，出力データを用紙に印刷する装置で，出力装置の代表的なものである。

普通，工学で使う入力，出力は，主としてエネルギーの入力，出力を指すが，電子計算機では，データあるいはプログラムが，入力したり出力したりする対象となる。したがって，入出力装置は，人間と電子計算機との間を結びつけるための装置であるといえる。

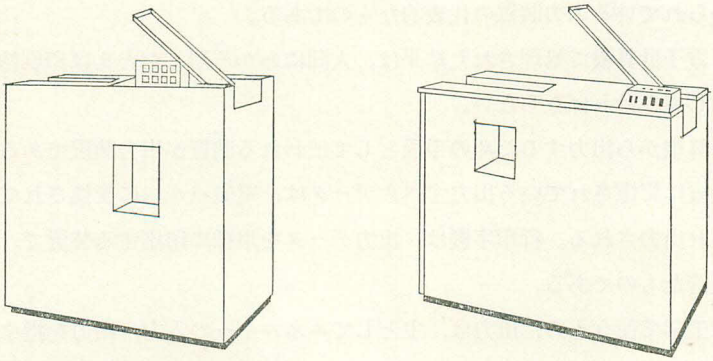
現在，電子計算機の高速化に対する努力が続けられているが，その場合のあい路となっているのが，これから説明しようとする入出力装置である。その理由は，中央処理装置の記憶，演算あるいは制御といった動作は，すべてマイクロ秒（10^{-6} s）～ナノ秒（10^{-9} s）といった電子的な速度であるのに対し，入出力装置の動作は，ほとんどがミリ秒（1 ms＝10^{-3} s）といった機械的な速度であるため，両者に千倍以上の速度差があり，全体的な処理速度が入出力装置の速度で支配されてしまうからである。したがって，入出力装置の高速化は，電子計算機の処理能力向上のための大きな課題と考えられている。

2.4.2 カード読み取り機[1]とカードせん孔機[2]

カードにせん孔（孔をあけること）されたデータを読み取る方法は，電子計算機が出現する以前から**パンチ・カード・システム**（せん孔カードを使って，事務処理を行なうシステム）において使用されていたが，電子計算機でもデータを入力するための方法として広く使われている。

カード読み取り機は，カードに記録されているデータを読み取り，中央処理装置へ送るために使われる装置であり，また，**カードせん孔機**は，中央処理装置で処理された結果をカードにせん孔するための装置である。第2.56図に，カード読み取

（1） card reader　（2） card punch

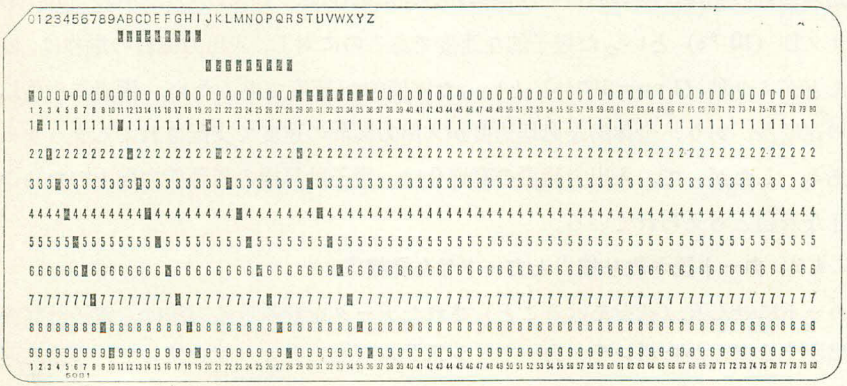


(a)　カード読み取り機　　(b)　カードせん孔機

第2.56図　外　観　図

り機とカードせん孔機の外観を示す。

1.　カード(1)　データの記録媒体として用いられる**カード**には，いろいろの形式のものがあるが，80けたカードがもっとも代表的である。



第2.57図　カードにせん孔されたデータ

(1)　card

このカードは，**第2.57図**に示すように，横方向は80けた，縦方向は12行からなっており，1けたで1文字を表わすようになっている。したがって，1枚のカードには最大80文字が記録される。

縦方向の行には，上から12（またはY），11（またはX），および0～9の番号がつけられている。

カードには，数字，英字，特殊記号（＋，－，％など）や，ときにはカナ文字も記録することができる。このうち，数字は0～9の位置にせん孔して，また，その他の字は，12，11，0の位置と1～9の位置にせん孔して表わされる。第2.57図は，数字（0～9）と英字（A～Z）がせん孔されている様子を示す。カードのせん孔に用いられるこの種のコード（符号）を，**ホレリス・コード**と呼んでいる。

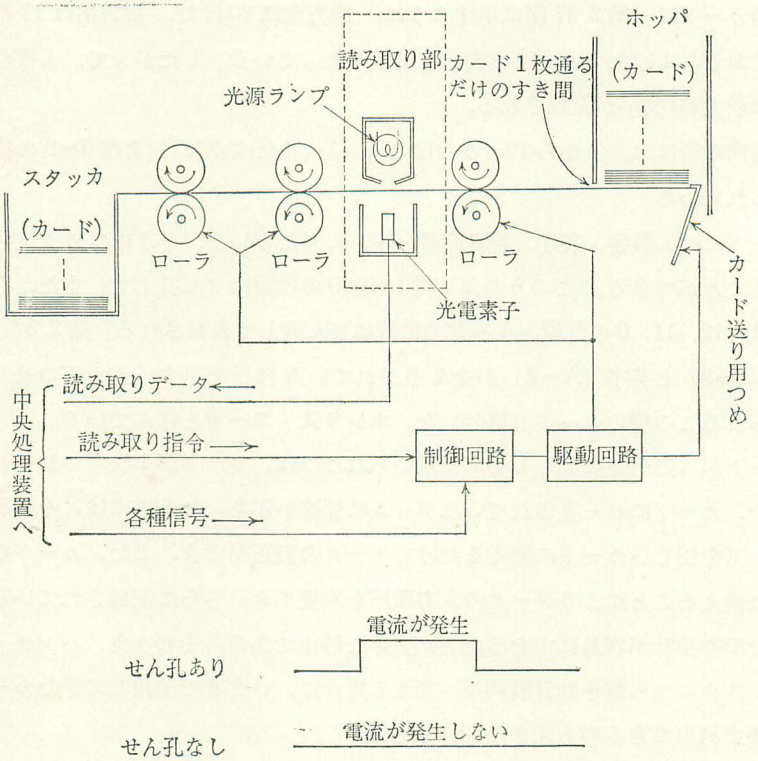
カードが入力記録媒体として広く用いられるのは，カードが1枚ずつ独立しているので，カードにせん孔されているデータに間違いがあったときには，その間違いのカードを正しいカードに変えるだけでデータの訂正ができ，また，カードの順序を入れ換えることによりデータの入力順序を変更でき，さらに記録されているデータの分類や集計が容易にできるなどが大きな理由である。そのうえ，パンチ・カード・システムから電子計算機利用へ変える場合に，いままで使用してきたカードをそのまま利用できるのも大きな利点である。

2. カード読み取り機　カード読み取り機には，各種方式のものがあるが，**第2.58図**は光電式のカード読み取り機の原理を示したものである。

せん孔ずみのカードは，まずホッパと呼ばれる場所に積み重ねられる。中央処理装置からカード読み取り指令がくると，ホッパの下部にあるカード送り用のつめが動いて，カード1枚を左へ押し出す。押し出されたカードは，ローラの回転によって左へ移動し，読み取り部を通ってさらに前進し，最後にスタッカと呼ばれる場所に積み重ねられる。

カードにせん孔されているデータは，カードが読み取り部を通過するときに読み取られる。

読み取り部は，光源ランプとフォトダイオードなどの光電素子（光を感知する素子）とで構成されていて，この両者の間をカードが通過するとき，せん孔の有無に



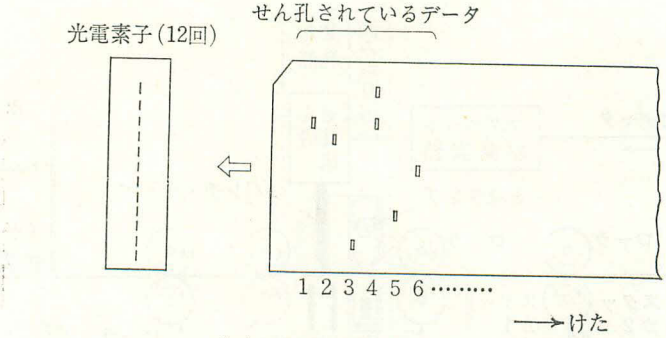
第2.58図　カード読み取り機の概略図

よって光電素子に投射されている光量が変化する（せん孔があれば光量は多く，せん孔がなければ光は光電素子に達しない）ので，これを電流の変化に変えて，読み取りが行なわれる。読み取られたデータは，中央処理装置で使用するいわゆる内部コードに変換されたうえで，記憶装置へ送られ記憶される。

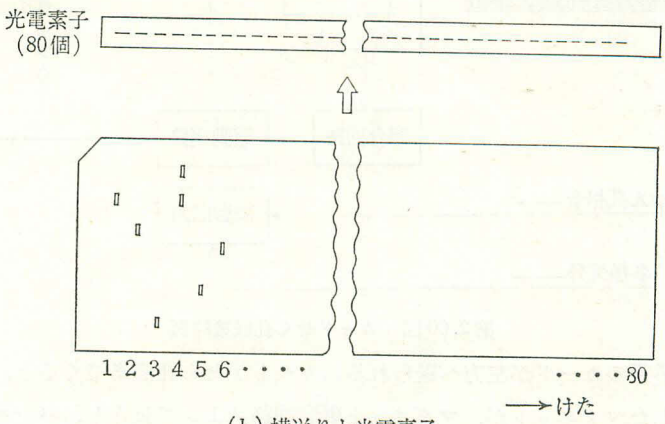
カードの送り方には**第2.59図**（**a**）のように，カードの長辺方向に送る**縦送り方式**(1)と，図(**b**)のように，短辺方向に送る**横送り方式**(2)とがある。

縦送りの装置の読み取り部では，光電素子が行の数（12個）だけでよいが，横送りの装置では，けたの数（80個）だけ必要である。また，前者は1けた目から順に読んでいくので，読み取り速度は遅いが，1けた読むたびに1字の内容がわかるの

（1）　column by column　　（2）　row by row



(a) 縦送りと光電素子



(b) 横送りと光電素子

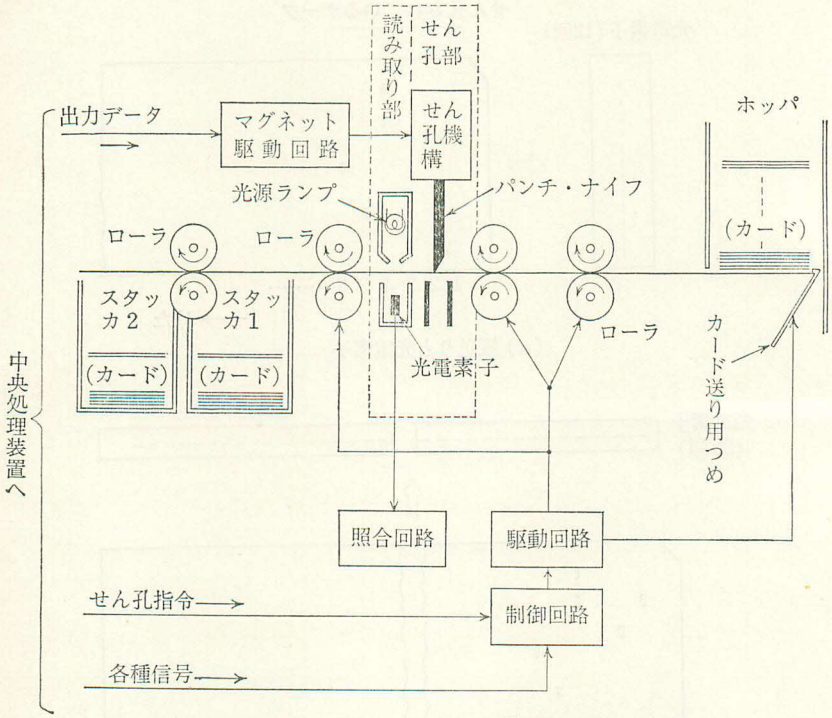
第2.59図 カードの読み取り方向

に対し，後者はカード1枚について12行読めばよいので読み取り速度は速いが，1行読んだだけでは意味がなく，1枚読んでから改めて1けた目からデータを組み立てる必要があり，そのための変換回路が必要となる。

現在用いられている光電式のカード読み取り機の読み取り速度は，数百～千枚/分程度である。

3. カードせん孔機 カードせん孔機は，中央処理装置で処理された結果をカードにせん孔するための装置で，**第2.60図**はその概略図である。

中央処理装置からせん孔指令が送られてくると，ホッパに積まれているカードの

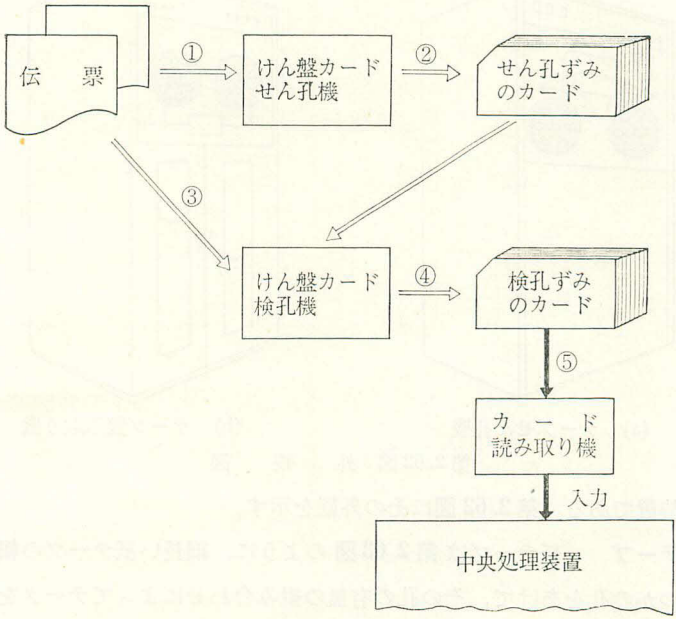


第2.60図　カードせん孔機概略図

中から一番下のカードが左方へ送られる。カードがせん孔部までくると，出力データに対応したマグネットが，マグネット駆動回路によって動作し，パンチ・ナイフでカードにせん孔する。せん孔されたカードの内容は，読み取り部で再び読み取られ，照合回路によって正しくせん孔されたかどうか検査される。検査を通過したカードはスタッカ1へ，また検査で誤りが発見されたカードはスタッカ2へ送られる。

カードせん孔機も，縦送りと横送りの2方式がある。現在用いられているカードせん孔機のせん孔速度は，100～300枚/分程度である。

4. **けん盤カードせん孔機とけん盤カード検孔機**　電子計算機にデータを入力するため，カードにそのデータをせん孔するには，人間（キー・パンチャ）が**けん盤カードせん孔機**のキーをたたいて行なう。けん盤カードせん孔機には，タイプライタに似たキーが配列してあって，キー・パンチャが伝票類の字を見てそのキーを



第2.61図　カード作成から入力までの例

たたくと，前述のホレリス・コードがカードにせん孔される。

これとは別に，**けん盤カード検孔機**が用意されていて，これにせん孔ずみのカードを入れ，別のキー・パンチャが，前に使ったと同じ伝票によってキーを操作し，正しくせん孔されているかどうかを検査する。さきのせん孔とこんどの検孔の結果が一致しないときは，間違いの表示が現われる。

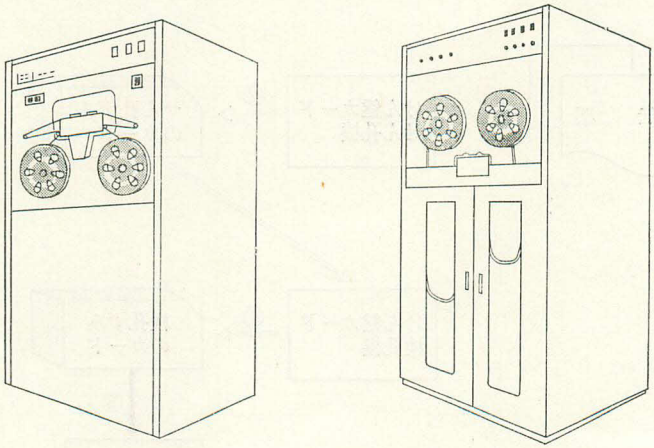
第2.61図は，カードのせん孔から入力までの過程を示したものである。

2.4.3　テープ読み取り機(1)とテープせん孔機(2)

紙テープにデータを記録する方法は，従来から電信の分野で広く用いられているが，電子計算機でもデータの入出力を行なう有力な手段の1つである。

テープ読み取り機は，紙テープに記録されているデータを読み取って入力するための装置であり，また，テープせん孔機は，処理された結果を紙テープにせん孔す

(1)　paper tape reader　　(2)　tape punch



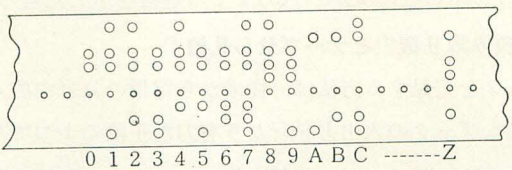
(a)　テープせん孔機　　(b)　テープ読み取り機

第2.62図　外　観　図

るための装置である。第2.62図にその外観を示す。

1.　紙テープ　紙テープは第2.63図のように，細長い紙テープの幅方向の1列にいくつかの孔をあけて，その孔の有無の組み合わせによってデータを記録するものである。テープには，数字（0～9），英字（A～Z），特殊な記号のほか，カナ文字やその他の符号が記録される。

いずれも1字分は，幅方向の1列のせん孔で表わされるが，1字が最大いくつの孔の組み合わせで表わされるかによって**5単位コード**，**6単位コード**，**7単位コード**および**8単位コード**がある。電子計算機では，主として6単位コードおよび8単位コードが用いられる。しかし，最近になって，これらの統一化の機運が高まり，**国際標準コード**（**ISO**(1)**コード**）が制定され，わが国でもこれに準拠した **JIS**(2)**コ**



第2.63図　紙テープ上のデータの記録8単位ＩＳＯコード

(1)　International Organization for Standardization の略

(2)　Japanese Industrial Standards の略

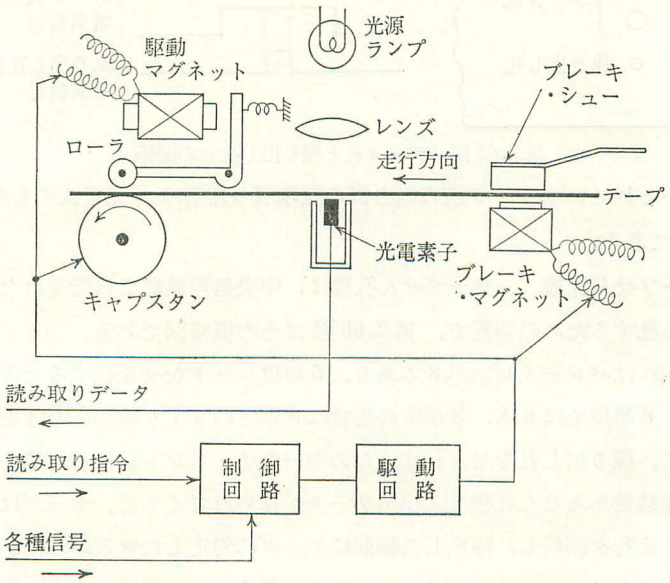
ードが制定されている。

第2.63図は，8単位コードで数字（0～9）および英字（A～Z）を表わした例である。紙テープには，データを表わす孔のほか，中央部に連続して**繰り出し孔**といわれる小さい孔もせん孔されている。

紙カードでは1枚に記録しうるデータが80けたまでの制約があるのに対し，紙テープは，記録の長さが自由であり，また，安価であるのが特徴である。

2. テープ読み取り機　テープ読み取り機には，各種の方式があるが，**第2.64図**で光電式のテープ読み取り機について説明する。

図で，キャプスタンは常時定速で回転しており，中央処理装置からテープ読み取り指令が送られてくると，駆動マグネットが働いてローラを下へ押しやるので，テープはキャプスタンに押しつけられ走行を開始する。テープの走行を止めるには，駆動マグネットの電流を切ると同時に，ブレーキ・マグネットを働かせると，ローラがばねの力ではね上がり，同時にブレーキ・シューでテープを押える。

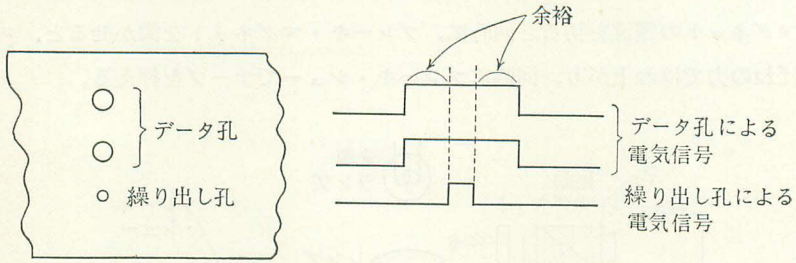


第2.64図　テープ読み取り機概略図

テープ読み取り機でも，テープの走行を正しく行なわせるために，磁気テープ装置の項で説明したような緩衝装置が使われている。

テープに記録されているデータは，テープが読み取り部を通過中に読み取られる。読み取り部は，光源ランプ，レンズおよび光電素子で構成されていて，テープにせん孔されている孔の有無によって生ずる光量の変化を電流の変化に変えて読み取りが行なわれる。読み取られたデータは，中央処理装置で使われる内部コードに変換されたうえで，記憶装置に記憶される。

データの孔が読み取られるとき，同列に並んでいる繰り出し孔も読み取られる。データ孔は，繰り出し孔より十分大きく，したがって**第2.65図**に示すように，データ孔により発生する電流波形は，繰り出し孔による電流波形より十分長いので，テープに多少のひずみがあっても，余裕を持って読み取りを行なうことができる。



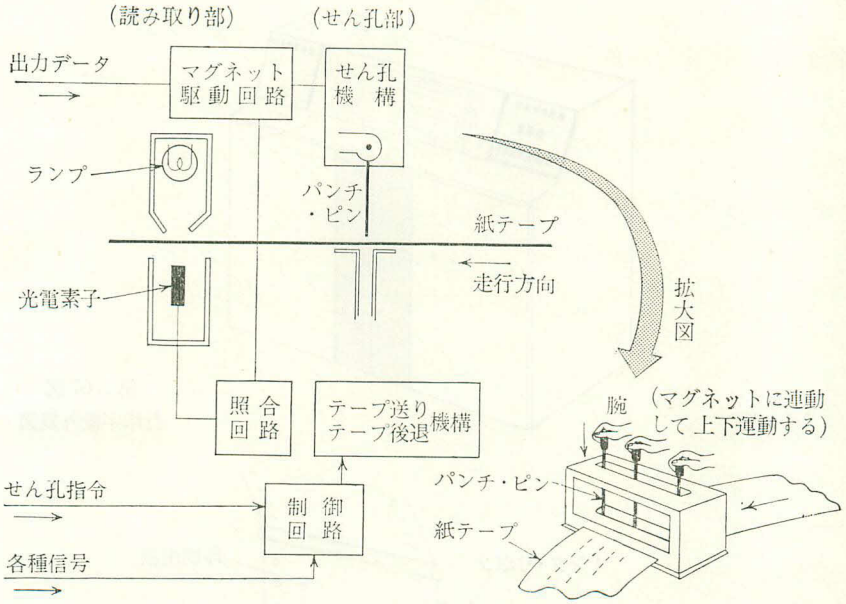
第2.65図　データ孔と繰り出し孔との関係

現在用いられているテープ読み取り機の読み取り速度は，光電式のもので1000字/秒程度である。

3.　テープせん孔機　テープせん孔機は，中央処理装置で処理された結果をテープにせん孔するための装置で，**第2.66図**はその概略図である。

せん孔部にはパンチ・ピンが8本あり，5単位コードをせん孔するときはそのうちの5本，6単位では6本，8単位の場合は8本というように使いわけをする。これとは別に，繰り出し孔をせん孔するためのパンチ・ピンも設けられている。

中央処理装置からせん孔指令と出力データが送られてくると，テープは1字分ずつ断続的に走行を開始し，停止した瞬間にデータに対応したマグネットが駆動回路によって動作し，パンチ・ピンでテープにせん孔する。繰り出し孔も，停止のたび



第2.66図 テープせん孔機概略図

にせん孔される。

せん孔されたデータは，読み取り部で再び読み取られ，正しくせん孔が行なわれたかどうかが調べられる。もし，せん孔に誤りがあれば，テープ後退機構でテープを後退させ，誤りのあった列に全部せん孔することにより誤りのデータをまっ消し，つぎのせん孔位置に改めてせん孔しなおす。

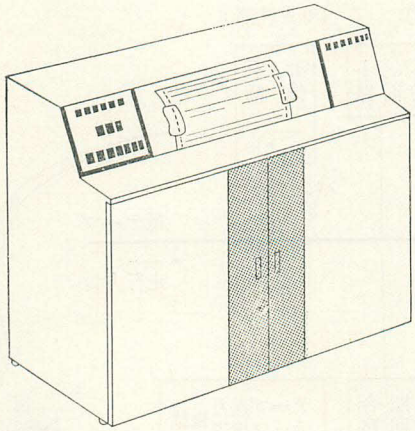
現在用いられているテープせん孔機の速度は，100～200字/分程度である。

このほか，人間（キー・パンチャ）がけん盤を押してテープにせん孔を行なう装置もあり，これを**けん盤テープせん孔機**と呼んでいる。

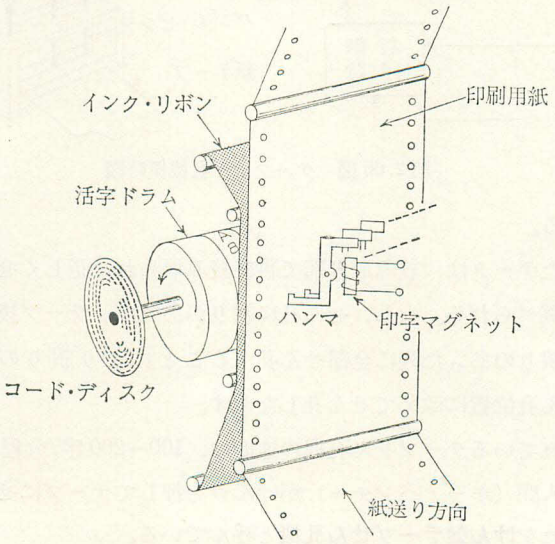
2.4.4 行印字機[1]

電子計算機で処理された結果を外部に出力する手段として，現在最も広く用いられているのは，印刷用紙に活字で印刷する方法である。しかし，通常のタイプライタ方式による印刷装置では，電子計算機の出力装置としてあまりにも速度が遅いの

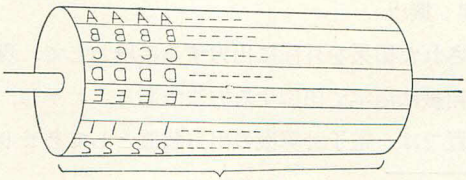
(1) line printer



第2.67図
行印字機外観図



第2.68図
印字機構



第2.69図
活字ドラム

で，種々の高速度印刷機が開発されている。

行印字機は，一度に1行ずつ印字することができるのでこの名がつけられた高速度の印刷機である。行印字機にも各種の方式のものがあるが，本項ではそのうち，ドラム式の行印字機について説明する。**第2.67図**にその外観を示す。

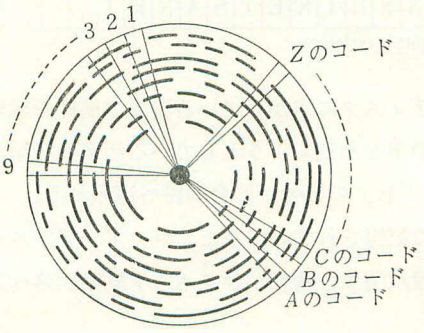
第2.68図は印字機構の概略図で，**第2.69図**はそのうち活字ドラムだけを取り出して示したものである。活字ドラムの表面には，出力データを印字するのに必要な字や記号の活字が配列されている。

さらに活字ドラムと対向して，1行の活字数と同数のハンマおよび印字マグネットが取りつけられている。普通，1行に印字される字数は120～132字である。

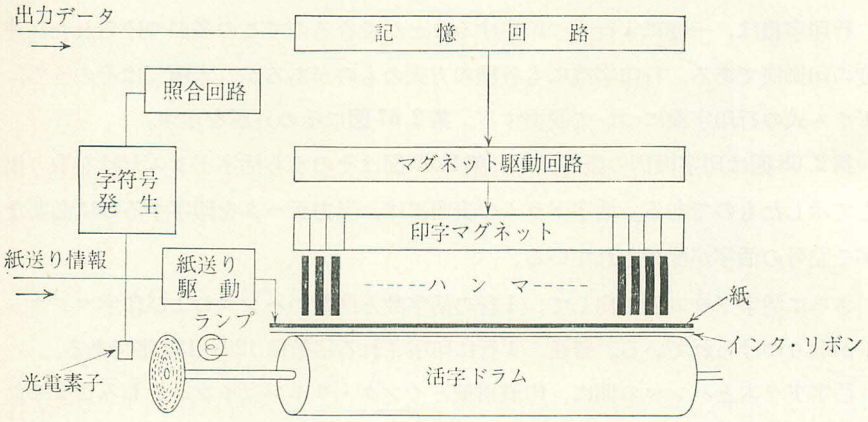
活字ドラムとハンマの間に，印刷用紙とインク・リボン（インクをしみこませた布）を通し，印字マグネットに電流を流すと，ハンマは用紙とインク・リボンを活字ドラムに押しつけるので，ハンマと対向した位置にある活字が用紙に印字される。

いま，どの活字がハンマと対向位置にきているかは，コード・ディスクによって識別する。これは**第2.70図**に示すように，プラスティック製の円板で，ドラムの活字配列と同順序に，それぞれの活字を表わす濃淡の模様が書かれている。これを活字ドラムに取りつけ，回転させながら光を投射すると，濃淡模様による光の変化が生ずるので，これを電流の変化に変えて活字の種類を識別することができる。

第2.71図は，行印字機の動作を示したものである。



第2.70図　コード・ディスクの一例（濃淡の模様でコードを表わす）

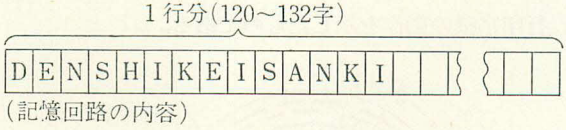


第2.71図　行印字機の動作概要

行印字機の活字ドラムは高速度で回転しており，これと同じ軸のコード・ディスクも，同じ速度で回転している。

行印字機により印刷される過程は，つぎのとおりである。

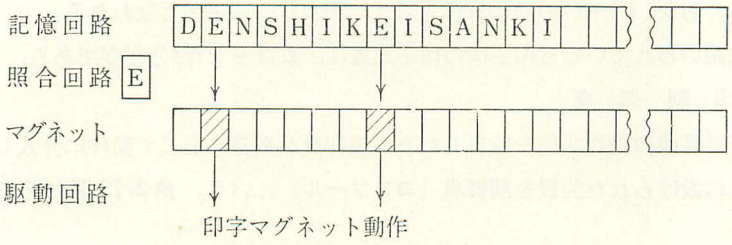
（1）　いま，中央処理装置から1行分の出力データ，たとえば「DENSHIKE ISANKI」という字（符号）が送られてくると，これらは行印字機にある記憶回路に，つぎのように一時的に記憶される。



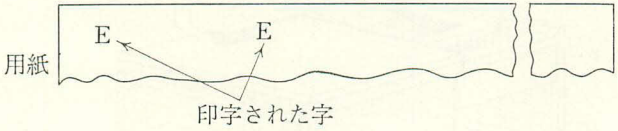
（2）　コード・ディスクに書かれている濃淡の模様を識別することによって，いままさにハンマの下を通過しようとしている活字の種類がわかる。それが「E」であったとすれば，「E」の符号が照合回路へ送られる。

（3）　記憶回路で記憶されている字と，コード・ディスクの字とが照合され，一致した文字の位置に関する情報がマグネット駆動回路へ送られて，印字マグネットを動作させる。

（4）　印字マグネットが動作すれば，ハンマが用紙とインク・リボンを活字ド



ラムに押しつけるので，ハンマの下を通過中の字「E」が用紙に印字される。

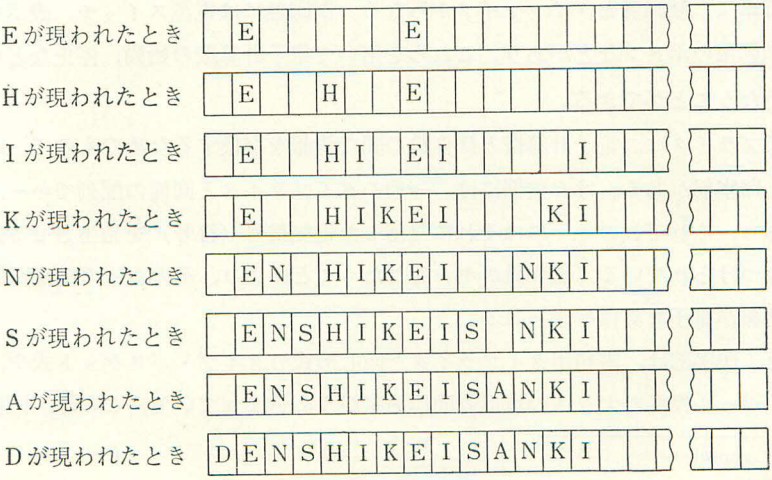


（5） 活字ドラムの回転により，つぎつぎと活字がハンマの下を通過するので，(2)～(4)の動作をくり返すことによって，活字ドラムが1回転する間に1行分の印字が行なわれることになる。たとえば，活字の現われる順序が，

E……H, I……K……N……S……Y, Z, A, ……D, E……

であれば，用紙の1行にはつぎのように順序印字されていく。

（6） 1行分の印字がすめば，中央処理装置からの指示によって紙送りが行な

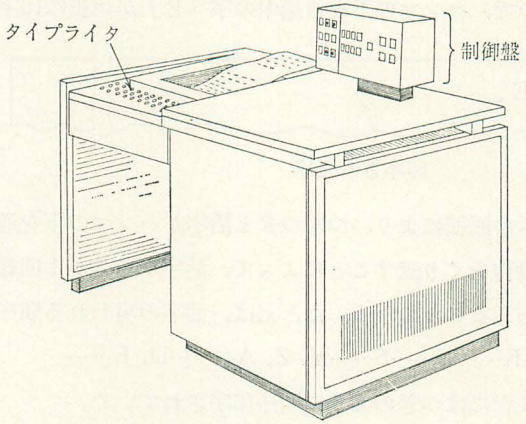


われ，再び(1)～(5)の動作によって，別の行の印字が行なわれる。

現在用いられている行印字機の印字速度は，数百～千行/分程度である。

2.4.5 制　御　卓[1]

電子計算機の動作状態を監視したり，操作員が必要に応じて動作に介入したりするために設けられた装置を**制御卓（コンソール）**という。**第2.72図**にその外観を示す。



第2.72図　制　御　卓

制御卓は，制御盤とタイプライタからなる。制御盤には電源スイッチ，表示ランプ，制御用のボタンなどがあり，これらを用いて電子計算機の始動，停止などの制御を行なうことができる。

タイプライタは，電子計算機と操作員の間で情報を交換するためのもので，けん盤部と印字部がある。けん盤部には，一般のタイプライタと同様の配列でキーが並んでおり，それぞれのキーにはそれに対応した電気信号（符号）を発生させる接点が取りつけられている。操作員がキーをたたくことにより，そのキーの接点が開閉し，情報が電子計算機へ送られる。

また，印字部は，事務用タイプライタと同じ形式のタイプ・バスケット式や，ゴルフ・ボール程度の大きさの球形表面に活字がうめ込まれていて，この球が回転し

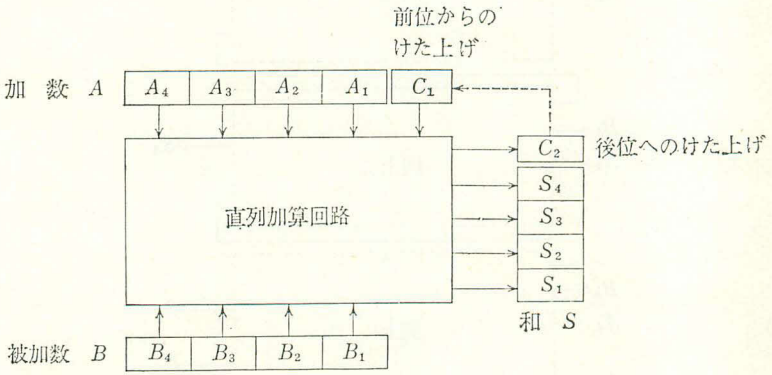
(1)　console

て活字を用紙に打ちつけるタイプ・ボール式など各種のものが用いられている。

2.5 演 算 装 置

電子計算機の演算装置は，その基本的な考え方においては卓上計算機の演算回路と同じものであるが，演算速度を上げるためのくふうや，四則演算のほかに論理演算を行なえるようになっている。

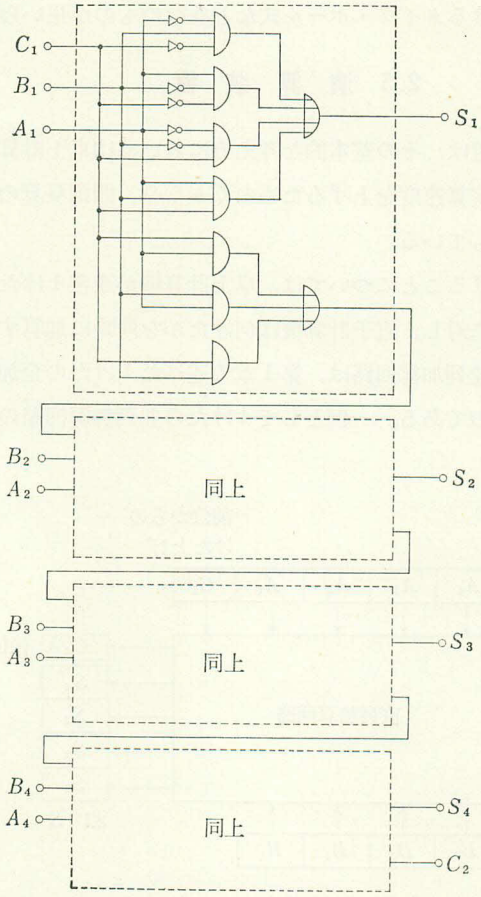
まず，演算速度を上げることについては，卓上計算機が通常1けたずつの**直列加算方式**を用いているのに対し，電子計算機は何けたかを同時に加算する**並列加算方式**を用いている。この並列加算回路は，第1章で述べた1けたの全加算器を必要な数だけ並列に並べたものである。一例として4けたの並列加算回路の構成を**第2.73図**に示す。



第2.73図 並列加算回路の構成

図において，C_1，C_2 のけた上げは，たとえば全部で8けたの加算を行なうときに，まず下4けたの並列加算を行ない，それから生ずるけた上げ C_2 を上4けたの並列加算時に加えてやるためのものである。並列加算回路の内部構成を**第2.74図**に示す。

4けたの並列加算回路を用いると，同じ4けたの加算を，直列加算回路で行なうのに比較して，演算時間は単純に考えて1/4になる。並列加算方式において，並列に何けたの演算を行なうかは，計算機の規模などによって異なるが，一般に4けた，



第2.74図　並列加算回路の内部構成

8けた，16けたなど4の倍数のけた数を用いている。これは，2進化10進法では10進1けたが4ビットで表わされているためである。

並列加算をより高速に行なうのに，下位からのけた上げについて考慮する必要がある。たとえば，つぎのような8けたの加算を並列加算で行なう場合を考える。

$$\begin{array}{r} 1\,1\,1\,1\,1\,1\,1\,1 \\ +)\ 0\,0\,0\,0\,0\,0\,0\,1 \\ \hline 1\,0\,0\,0\,0\,0\,0\,0\,0 \end{array}$$

この場合，最上位けたの加算はすぐには行なうことができず，最下位けたから生ずるけた上げが順に上位へ上がり演算が行なわれ，けた上げが最上位けたまで到達したときに，はじめて最上位けたの演算が行なわれることになる。この下位からのけた上げが最上位けたまで到達するのに要する時間は，並列のけた数が多くなればなるほど長くなり，演算速度を重視する電子計算では無視できなくなる。

この問題を解決するために，**けた上げ先取り方式**という方法を取る場合がある。この方式は，あるけたに下位からのけた上げがあるか否かを，下位から順に演算することなしに調べるものである。例として，さきの8けたの計算をつぎのように上，下各4けたに分けて行なう場合を考える。

```
←上→ ←下→
 1111 | 1111 (被加数)
+)0000 | 0001 (加　数)
-------+------
10000 | 0000
```

この場合，下4けたからのけた上げがあるか否かがわかれば，上4けたは下4けたと同時に加算することができる。順に加算することなしに，下4けたからけた上げがあるか否かを判定するには，つぎのように行なえばよい。

① 調べようとするけたの中に，加数，被加数とも1（これを and がとれるという）のけたがあるか否かを調べる。

② もし，and がとれるけたがある場合には，そのけたから上位のけたが，加数，被加数のいずれかが1（これを or が取れるという）になっているか否か（加数，被加数とも0になっていないかどうか）を調べる。

この2つの条件が満足されている場合には，上位のけたへのけた上げを生ずる。これを上の例で見ると，最下位のけたが加数，被加数とも1になっており，その上位けたはいずれも被加数が1になっているため，この下4けたからはけた上げがあることになる。またつぎに示す2例も，いずれもけた上げの条件を満足しており，けた上げを生じている。

```
 0101        0110
+)1011      +)1010
-------     -------
10000       10000
```

電子計算機の演算装置では，純2進演算，2進化10進演算のどちらでもできるように回路が構成されている。また，数値データについての演算だけでなく，数値以外の各種データについて，突き合わせ，比較判断などの処理を行なわせるために，and，or の論理演算もできるようになっている。

これらの2進演算，2進化10進演算，論理演算などは，それぞれ別個の演算回路によって行なわれるのではなく，1つの演算装置が制御装置からの制御信号によって，あるときは2進演算を，また，あるときは論理演算をというように切り換えられ，多能的に使用されている。

2.6 制御装置

2.6.1 概要

人間が卓上計算機を使って計算をする場合，どのような計算をし，得られた結果をどのように判断し，そのつぎはどういう計算をするかなどの手順は，すべて人間の頭の中できちんと組み立てられており，人間はこの手順にしたがい種々の判断をしながら卓上計算機を操作して計算を進めていく。結局，計算に必要な制御は，すべて人間の判断にまかされているものと考えることができる。

2.1でも述べたように，電子計算機では，卓上計算機の場合には人間の判断にまかされていた制御の部分まで含めて，自動的に計算を行なうことができる。この点が，電子計算機と卓上計算機の最も大きく，根本的な相違点であるといえる。

本節では，こうした制御がどのようにして行なわれるかについて説明する。

2.6.2 プログラム

たとえば，数の和を求める計算を電子計算機で行なわせる場合の手順は，つぎのようになる。

① 最初の数字をカードから読む。
② つぎの数字をカードから読む。
③ 2数を加算する。
④ 計算結果を印刷する。

このような手順（命令）を電子計算機に覚え込ませておくとともに，数字を記憶

した紙カードを用意すれば，電子計算機は①から④までの一連の命令を1つずつ忠実に実行し，計算処理を進めていく。そこには人間が介在する余地，あるいは必要性はまったくなく，電子計算機はあらかじめ人間が与えたプログラムで指示されている通り，順序正しく，しかも非常な高速度でデータの処理を行なう。

ところで，われわれがプログラムを作る，すなわち，プログラミングを行なう場合には，**プログラム言語**(1)という特殊な言語を使用する。プログラム言語には，**機械語**(2)，**アセンブリ語**(3)および**コンパイラ語**(4)などいくつかの体系がある。

機械語というのは，2進符号あるいは2進符号に容易に変換しうる数字を主体としてすべての命令を書き表わすもので，当然暗号のような言語となり，また，各命令を納める記憶装置のアドレスをいちいち指定する必要があるため，熟練したプログラマでも機械語でプログラムを書くのには，きわめて長い時間を要する。したがって，機械語は電子計算機の記憶容量，処理時間をもっとも効率よく使用するなど，きわめて限定された目的以外には，プログラム言語としてはほとんど使用されない。

このような機械語の不便さを解消する目的で考察されたのが，日常語に近い記号で命令を書くことのできる（たとえば，加算であれば「ADD」の「A」を書く）アセンブリ語である。アセンブリ語では，プログラムを格納するアドレスを指定しなくても電子計算機が適当に決めてくれるので，プログラムを作成する時間も短縮され，プログラムを作成する際の誤りも大幅に減少し，また，プログラムの修正も容易となった。

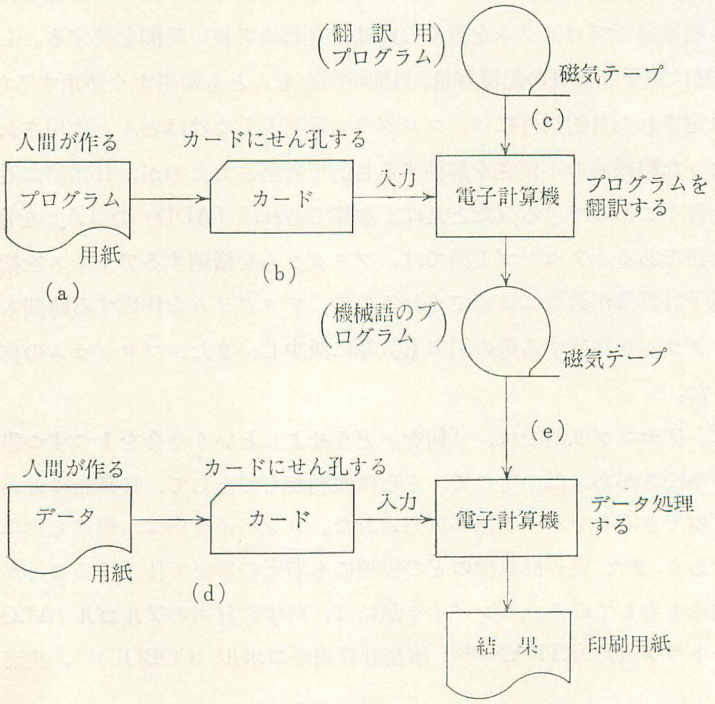
しかし，アセンブリ語では，「何を，どうせよ」という命令を1つずつ書かねばならない不便さがあった。そこで，さらに便利なものとして，日常語に近い文章で書くことのできるコンパイラ語が開発された。コンパイラ語は，規則も簡単で修得が容易であり，また，電子計算機のどの機種にも若干の修正で使用することができるという利点を有している。コンパイラ語には，科学計算用の**アルゴル** (ALGOL)(5)や**フォートラン** (FORTRAN)(6)，事務計算用の**コボル** (COBOL)(7)，両者の機能

（1） programming language （2） machine language （3） assembly language
（4） compiler language （5） Algorithmic Language の略
（6） Formula Translator の略 （7） Common Business Oriented Language の略

を併せ持った **PL/I** などがある。

ところで，電子計算機はデータ処理に際しては，機械語のプログラムしか命令として受けつけてくれない。したがって，アセンブリ語やコンパイラ語で作られているプログラムは，データ処理を行なう前に機械語のプログラムに変換する必要がある。この変換は，**翻訳**とも呼ばれ，電子計算機を使って行なわれる。

たとえば，アセンブリ語で作られているプログラムは，**アセンブラ**という翻訳用プログラムを使って，また，コボルで作られているプログラムは，**コボル・コンパイラ**という翻訳プログラムで，フォートランは**フォートラン・コンパイラ**で，それぞれ機械語のプログラムに翻訳される。



図では，2台の電子計算機を使うように見えるが，1台の電子計算機をプログラムの翻訳と，データの処理に時間をずらして使うのである。

第2.75図　プログラムの翻訳と実行

こうして電子計算機によりデータが処理される手順を**第2.75図**により整理すると，つぎのようになる。

① アセンブリ語やコンパイラ語でプログラムを作る。

② プログラムをカードにせん孔し，電子計算機に入力する。

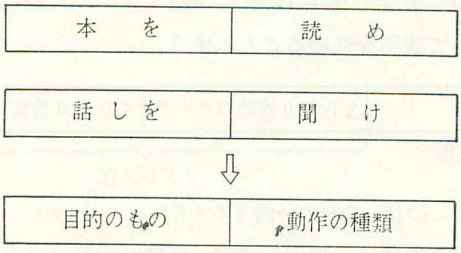
③ 電子計算機は，磁気テープに記憶しておいた翻訳用プログラムを使って，機械語のプログラムに翻訳し，別の磁気テープに記録する。

④ 処理したいデータをカードにせん孔する。

⑤ 機械語のプログラムが磁気テープより読み出され，そのプログラムによりデータが入力され，処理され，そして結果が出力される。

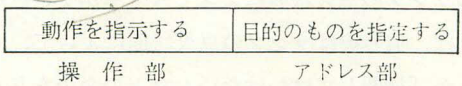
2.6.3 機械語の命令

人間が人間に与える命令は，たとえば「本を読め」とか，「話しを聞け」などのように，「何を，どうせよ」の形をとることが多く，**第2.76図**のように「目的のもの（本とか話し）」と，「動作の種類（読めとか聞け）を表わすもの」とで組み立てられている。



第2.76図 人間に与える命令の形

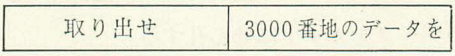
機械語の命令も，これとよく似た形で構成されている。その形は**第2.77図**のように，「動作を指示する部分」と，「目的のものを指定する部分」からなり，前者を**操作部**(1)，後者を**アドレス部**(2)呼んでいる。



第2.77図 計算機に与える命令の形式（1アドレス・コード）

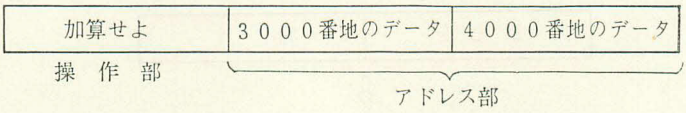
(1) operation part (2) address part

操作部では，「加算せよ」，「減算せよ」，「入力せよ」，「出力せよ」など，電子計算機に行なわせようとする動作の種類を指示し，アドレス部では，処理の対象となるデータが記憶されている記憶装置のアドレスを指定する。たとえば，「記憶装置の3000番地に記憶されているデータを取り出せ」という命令ならば，**第2.78図**のようになる。



第2.78図　命令の一例（1アドレス・コード）

このように1個のアドレス部を持つ命令を，1アドレス・コードの命令という。機械語の命令には，1アドレス・コードのほかに，電子計算機の機種によってはアドレス部を2つ持つもの（2アドレス・コードの命令という），3つ持つもの（3アドレス・コードの命令という）など，各種の形式の命令があるが，ここでは，2アドレス・コードによる命令の一例として「3000番地のデータと，4000番地のデータとを加算せよ」という命令を**第2.79図**に例示するにとどめ，以後は1アドレス・コードの命令について説明を進めることにする。



第2.79図　命令の一例（2アドレス・コード）

前述のように電子計算機では，数字，文字，特殊な記号（たとえば＋，－，×など）は，すべて0と1の組み合わせ，すなわち，2進数で表現される。これと同様に，機械語の命令も2進数で表わされている。

2進数で命令するということを身近かな例で考えてみよう。交さ点には信号機が設備されており，そのランプの点火状態によって人や車が交さ点を渡っている。

これは見方を変えれば，信号機はランプの点火状態によって，人や車に対して「横断してもよい」とか「横断してはいけない」などの命令を与えているものと考えることができる。

いま，ランプが点火している状態を1，消ている状態を0とすれば，

青 黄 赤

「赤ランプ点火」……[0 0 1]（命令）…「横断してはいけない」

「黄ランプ点火」……[0 1 0]（命令）…「注意せよ」

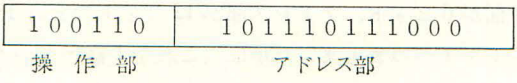
「青ランプ点火」……[1 0 0]（命令）…「横断してもよい」

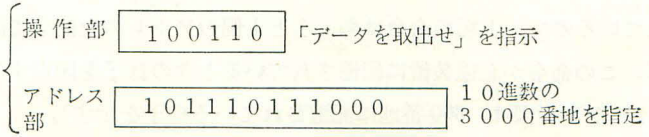
というように，0と1の組み合わせで3種類の命令が出されており，これはとりもなおさず，2進数を使った命令とみなすことができる。

再び電子計算機に話しをもどすと，電子計算機が実行する動作にはいろいろあるが，たとえば「加算せよ」を [100000]，「減算せよ」を [100001]，「乗算せよ」を [100011] というように，動作の種類に応じて一定形式の2進数を対応させれば2進数で動作の種類を指定することができる。このような2進数を命令の操作部の内容とすればよい。

一方，記憶装置にはアドレスがつけられているので，このアドレスを2進数で表わし，命令のアドレス部とすれば，その指定によって何番地のデータの処理をすればよいかが明らかとなる。

このようにして作られた命令の一例が**第2.80図**に示されている。操作部には，[100110] の2進数が記憶されている。いま，これを「データを取り出せ」という動作を指示するものとする。また，アドレス部には [101110111000] の2進数が記



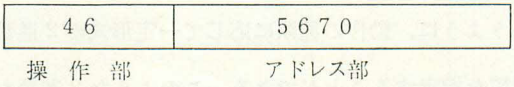


第2.80図 機械語の命令の一例（2進数で表示したとき）

録されているが，これは記憶装置のアドレスを示すもので，10進数でいえば，3000番地である。したがって，図の命令は，「3000番地のデータを取り出せ」ということになる。

電子計算機の制御装置は，この命令の操作部の内容から，「データを取り出す」動作をすればよいことを知り，また，アドレス部の内容から，「3000番地」に記憶された内容を処理すべきことを知るので，記憶装置に対して，そのために必要な制御の指令を与えることになる。

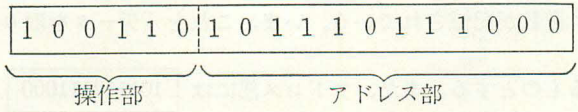
このように，命令はすべて2進数で作られるが，けた数が多いため人間が見たときわかりにくい。しかし，この命令を8進数で表示すると，**第2.81図**のようになって，かなり扱いやすくなる。



第2.81図　第2.80図の命令を8進数で表示したとき

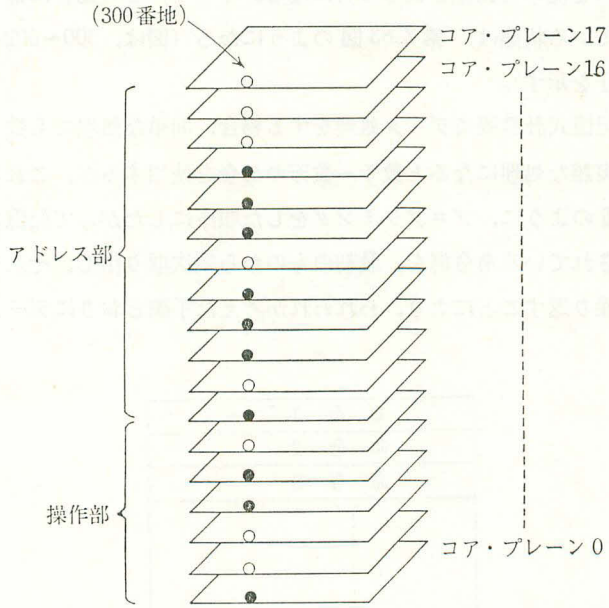
2.6.4 命令の記憶

さきに述べたように，プログラム記憶式計算機は，処理に先だって処理手順を示すプログラムを記憶しておく必要があり，記憶装置には前項で述べたような形式の命令が多数記憶されている。ところで，前述の

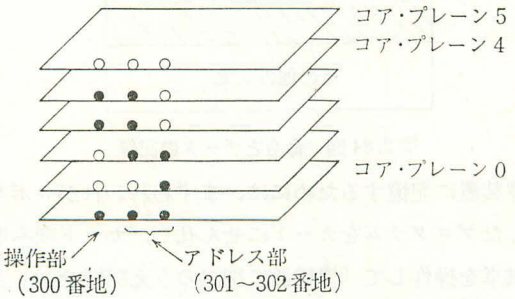


という命令は，操作部が6ビット，アドレス部が12ビットの合計18ビットで構成されている。18ビットを1つのまとまった単位（これを**1語**(1)という）として取り扱う電子計算機の記憶装置では，18ビット，すなわち，1語ごとに1個のアドレスが与えられているので，上記の命令はちょうど1個のアドレスを使って記憶されることになる。この命令が記憶装置に記憶されているときの様子を図示すると，**第2.82図**のようになる（図は，300番地に記憶されている様子を示す）。

(1)　word



●は“1”，○は“0”を記録していることを示す。
コア・プレーン0から上方向にみること
第2.82図　命令の記録（1語ビットで命令が構成されている）



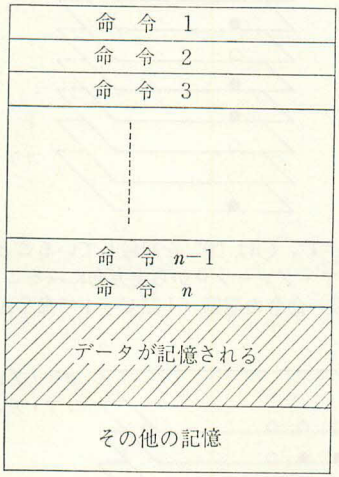
●は“1”，○は“0”を記録している。コア・プレーン0から上方向をみる。

第2.83図　命令の記録（1字6ビットで3字で命令が構成されている）

これに対し，1字6ビットを単位として扱う電子計算機の記憶装置は，6ビットすなわち，1字ごとに1個のアドレスが与えられているので，上記命令は3個の連

続したアドレスを使って記憶されることになる。そして，この場合に命令が記憶装置に記憶されている状態は，**第2.83図**のようになる（図は，300～302番地に記憶されている様子を示す）。

プログラム記憶式計算機でデータ処理をする場合，簡単な処理でも数十の命令が必要であり，複雑な処理になると数千～数万の命令を使用するが，これらの命令群は，**第2.84図**のように，プログラミングをした順序にしたがって記憶される。このように記憶されている命令群を，最初のものから順次取り出し，それを実行するという過程を繰り返すことにより，われわれが考えた手順どおりにデータの処理が進行する。



第2.84図　命令とデータの記録

プログラムを記憶装置に記憶するためには，まずわれわれがコボル，フォートランなどにより記述したプログラムをカードにせん孔し，カード読み取り機におき，一方，操作員が制御卓を操作して「機械語に翻訳のうえ記憶せよ」という指示を与えると，われわれが作ったプログラムは，べつべつに用意されている翻訳用プログラムの働きにより機械語の命令に翻訳され，記憶装置に記憶される。

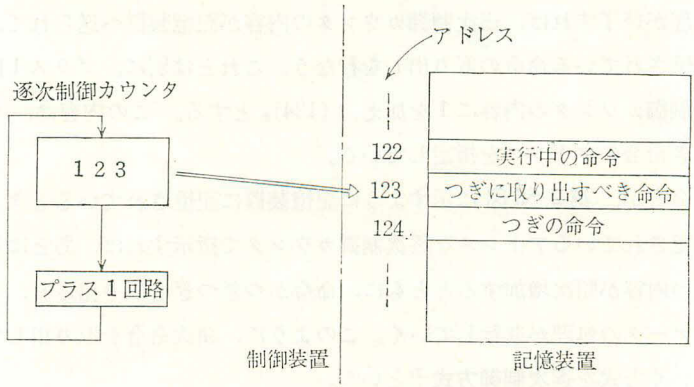
なお，アセンブリ語のプログラムを機械語に翻訳することを**アセンブルする**といい，コンパイラ語のプログラムを翻訳することを**コンパイルする**という。また，機

械語の命令を記憶装置に記憶することをプログラムの**ロード**(1)という。

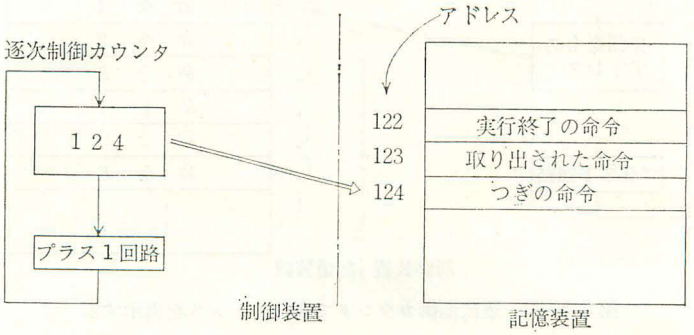
2.6.5 命令の取り出しと実行

プログラム記憶式計算機では，あらかじめ記憶装置に記憶されている命令によりデータの処理を行なうが，それには記憶装置に記憶されている命令を取り出す**命令取り出し段階**(2)と，取り出した命令について，何をする命令であるかを調べて必要な制御を行なう**命令実行段階**(3)の2つの段階がある。

1. **命令取り出し段階**　電子計算機が，あらかじめ定められた命令を逐次実行



（a）122番地の命令実行中



（b）123番地の命令が取り出されたとき

第2.85図　命令の取り出し

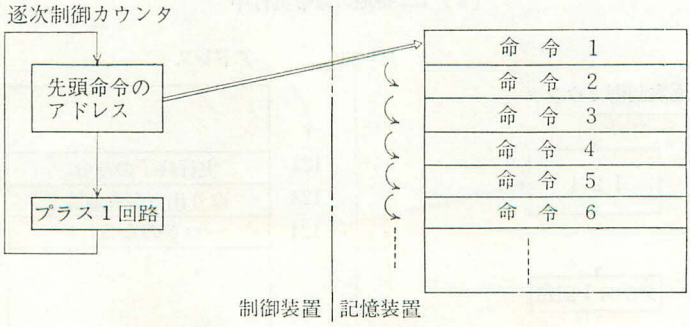
(1) to load　(2) fetch cycle　(3) execution cycle

するためには，まず，最初の命令が記憶されている記憶装置のアドレスを知り，それからつぎつぎと記憶装置から命令を取り出してくる必要がある。この目的のために，制御装置内に**逐次制御カウンタ**(1)といわれるカウンタが設けられている。

第2.85図に示すように，一連の命令が記憶装置に連続して記憶されている。

いま，$(122)_8$番地（以下8進数で表示することにする）の命令で，ある処理を実行しているときには，逐次制御カウンタには，つぎに取り出して実行すべき命令が記憶されているアドレス，すなわち，$(123)_8$が記録されている。そして，現在実行中の処理が終了すれば，逐次制御カウンタの内容が記憶装置へ送られて，$(123)_8$番地に記憶されている命令の取り出しを行なう。これとは別に，プラス1回路により，逐次制御カウンタの内容に1を加え，$(124)_8$とする。この内容は，つぎに取り出すべき命令のアドレスを指定している。

一連の命令が，**第2.86図**に示すように記憶装置に記憶されているとき，最初の命令が記憶されているアドレスを逐次制御カウンタで指示すれば，あとは逐次制御カウンタの内容が順次増加するとともに，命令がつぎつぎと取り出され，その命令によってデータの処理が進行していく。このように，順次命令を取り出して制御を行なっていく方式を**逐次制御方式**(2)という。



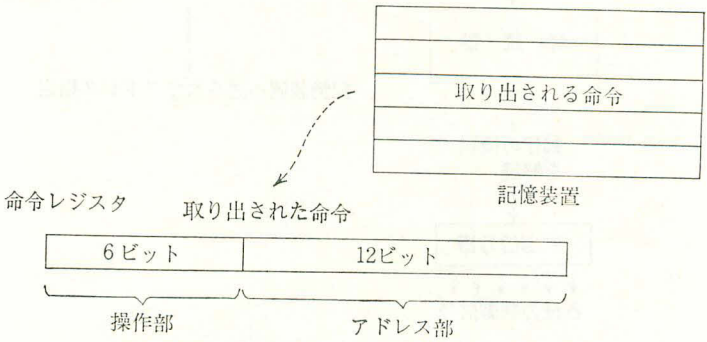
第2.86図　逐次制御カウンタで順次アドレスを指示する

先頭の命令が記憶されているアドレスは，別のプログラムで指定される場合もあり，また，電子計算機が自動的に指定して，その値を逐次制御カウンタに入れる場

(1)　sequence control counter　　(2)　sequence control system

合もある。

こうして取り出された命令は，**第2.87図**のように制御装置内の**命令レジスタ**(1)に，一時的に記憶される。



第2.87図　命令レジスタに取り出された命令の形式

前述のように，命令は操作部とアドレス部で構成されており，取り出された命令もこの形のまま命令レジスタに記憶される。したがって，命令レジスタの上位（左側）6ビットは操作部であり，また，下位（右側）12ビットはアドレス部である。

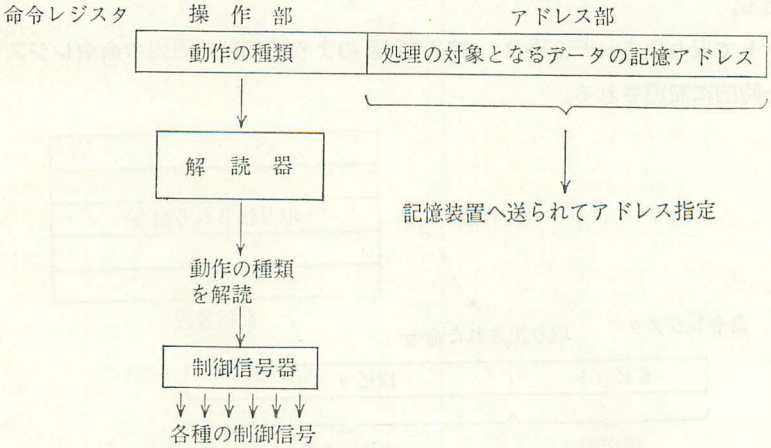
2. 命令実行段階　命令レジスタに記憶された命令により制御を行なうには，まず，命令の操作部を調べて，それが何の命令であるかを知る必要がある。命令レジスタのうち，上位6ビットの操作部が解読器に送られると，解読器はその内容から動作の種類を解読し，それを制御信号器に通知する。制御信号器は，通知された動作を実行できるように各種の制御信号を発生する。

一方，命令レジスタの下12ビットの内容は，処理の対象となるデータの記憶アドレスを指定しており，その内容はアドレス情報として記憶装置へ送られる（**第2.88図**参照）。

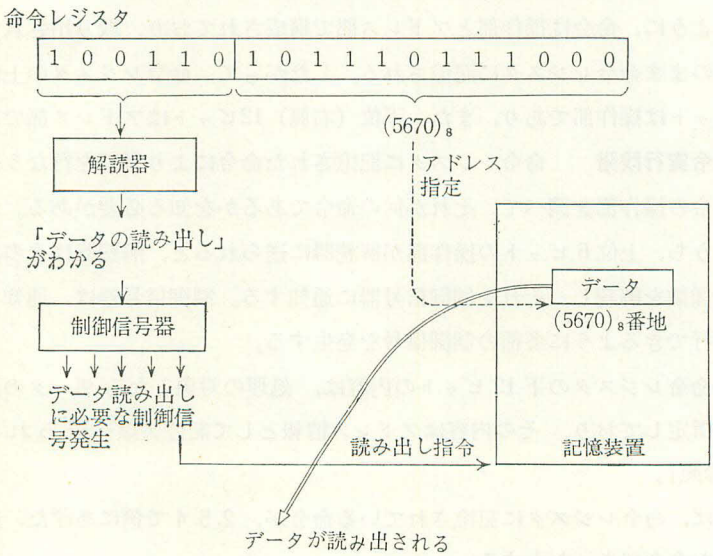
たとえば，命令レジスタに記憶されている命令が，2.5.4で例にあげた，**第2.89図**のような命令であったとする。

まず，命令レジスタの操作部を示す上位6ビットが解読器に送られ，その2進数から「データの読み出し」を行なうべきことがわかるので，それを制御信号器へ通

(1)　instruction register：略して IR



第2.88図　制御の実行



第2.89図　制御の実行例

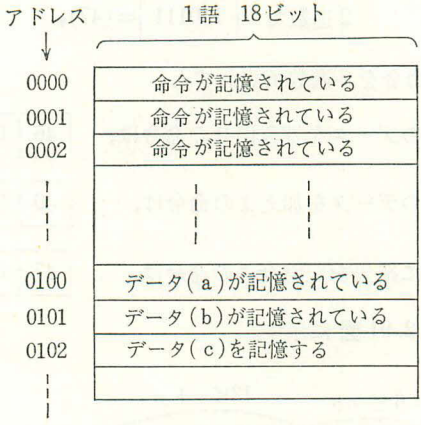
知する。同時に命令レジスタの下位12ビットのアドレス部が，記憶装置へ送られ，指定されたアドレスに記憶されているデータが読み出される。

この例は，単に記憶装置にあるデータを読み出すだけの動作を行なうものである

が，命令の種類によって，たとえば加算器を働かせたり，入力装置を働かせてデータを入力したり，あるいは出力装置で出力したりなど，さまざまの動作の制御が行なわれる。

2.6.6 制 御 の 方 法

1. 仮定の設定 $a+b=c$ という簡単な計算を行なう場合について，制御の方法を説明する。ここで，つぎのようないくつかの仮定をおく(**第 2.90 図**参照)。



第 2.90 図 記憶内容

(1) 特に断わらない限り 8 進数で表示し，( )$_8$ で表わす。

(2) 記憶装置は，1 語 18 ビット単位に $(0000)_8$ から順にアドレスをつける。

(3) プログラムの各命令は，1 語 18 ビットの 1 アドレス・コードで構成されており，$(0000)_8$ 番地からアドレスの順に記憶されている。

(4) データは，それぞれ 18 ビットで表わされており，データ (a) は $(0100)_8$ 番地に，データ (b) は $(0101)_8$ 番地に記憶されている。また，結果のデータ (c) は，$(0102)_8$ 番地に記憶するものとする。

データ処理を行なうために必要な手順は，

① $(0100)_8$ 番地に記憶されているデータを読み出す。

② $(0101)_8$ 番地のデータを加えて，

③ $(0102)_8$ 番地に結果を記憶する。

すなわち，動作の種類としては，「データを読み出す」，「加算する」，「記憶する」の3種類を使用すればよい。これらの動作を命令の操作部で指定する場合，つぎのような符号を用いるものとする。

データを読み出す	2進数で	100110	$=(46)_8$
加算する	2進数で	100000	$=(40)_8$
記憶する	2進数で	100111	$=(47)_8$

したがって，上記各命令を8進数で表わすと，

①の $(0100)_8$ 番地のデータを読み出せの命令は，

46	0100

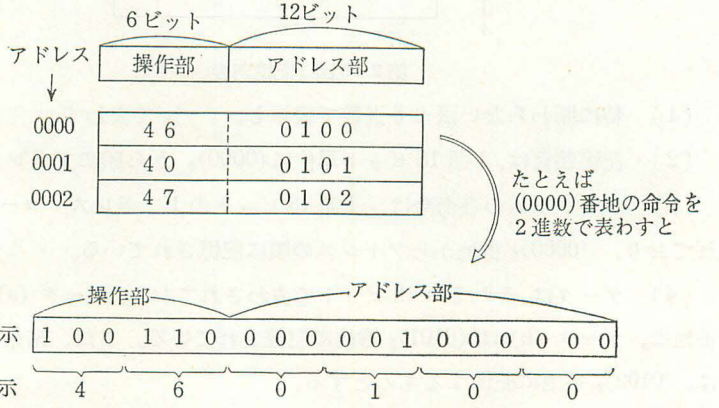
②の $(0101)_8$ 番地のデータを加えよの命令は，

40	0101

③の $(0102)_8$ 番地に結果を記憶せよの命令は，

47	0102

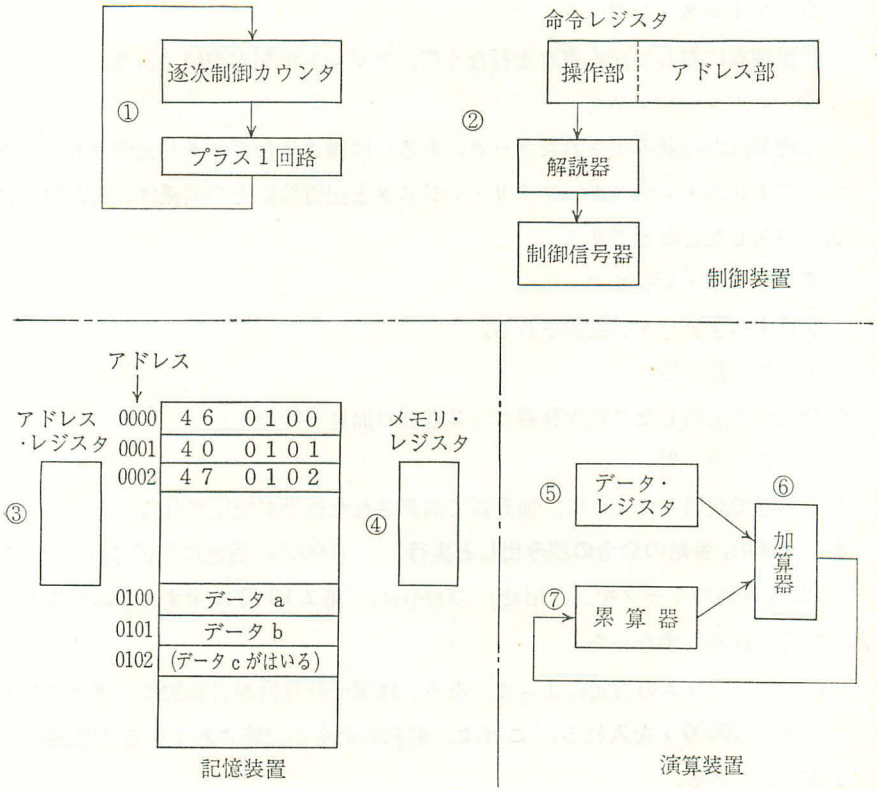
となる。この様子を**第2.91図**に示す。



第2.91図　命令の記憶

2.　装置類の動作　　**第2.92図**は，上記命令を実行するに際しての，各装置の動作を示したものである。

①　逐次制御カウンタ（SCC）と，プラス1回路



第2.92図 各装置の動作

逐次制御カウンタは，前述のように，実行中の命令のつぎの命令が記憶されている記憶装置のアドレスを指定する。命令の取り出しが終わると，プラス1回路によって，その内容は自動的に1ずつ加算されていく。

② 命令レジスタ，解読器，制御信号器

命令レジスタは，読み出された命令が一時的に記憶されるレジスタである。いま考えている例では，操作部は6ビット，アドレス部は12ビットで構成されている。

解読器は，操作部の内容によって「動作の種類」を解読する。また，制御信号器は，「動作の種類」に応じて，それを行なうのに必要な制御信号を発生する。

③ アドレス・レジスタ

記憶装置に対して読み書きを行なう際，アドレス情報が記憶される。

④ メモリ・レジスタ

記憶装置から読み出されたデータ，あるいは書き込むデータが記憶される。なお，アドレス・レジスタ，メモリ・レジスタと記憶装置との関係は，第2.24図により説明したとおりである。

⑤ データ・レジスタ

演算すべきデータが記憶される。

⑥ 加 算 器

演算装置で説明した2進加算器で，2進数の加算を行なう。

⑦ 累 算 器

演算装置で説明したように，加算器で演算された結果が記憶される。

3. $(0000)_8$ 番地の命令の読み出しと実行 $(0000)_8$ 番地に記憶されている「$(0100)_8$ 番地のデータを読み出せ」の命令は，**第2.93図**に示すように読み出され，実行される。すなわち，

① プログラムの指定によって，あるいは電子計算機が自動的に，逐次制御カウンタへ $(0000)_8$ を入れる。これは，最初の命令が記憶されている記憶場所のアドレスである。

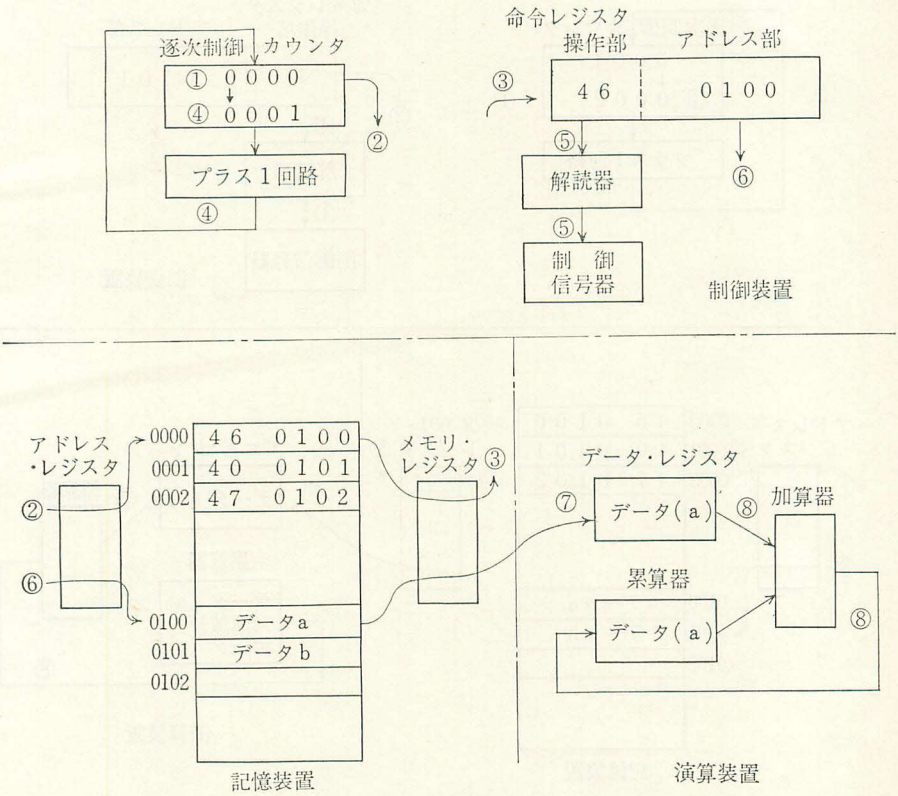
② 逐次制御カウンタの内容がアドレス・レジスタへ送られ，同時に記憶装置に対して，読み出し指令を出す。

③ 最初の命令がメモリ・レジスタに読み出され，命令レジスタへ送られる。

④ 逐次制御カウンタの内容に1が加えられ，$(0001)_8$ となる。これは，いまから実行しようとする命令のつぎの命令が記憶されている記憶装置のアドレスである。

⑤ 命令レジスタに記憶された命令のうち，上位6ビット $(46)_8$ が解読器へ送られて解読される。その結果，「データ読み出し」動作を行なうべきことがわかるので，制御信号器からその動作を行なうに必要な制御信号が出される。

⑥ ついで，命令のアドレス部の内容 $(0100)_8$ がアドレス・レジスタへ送ら



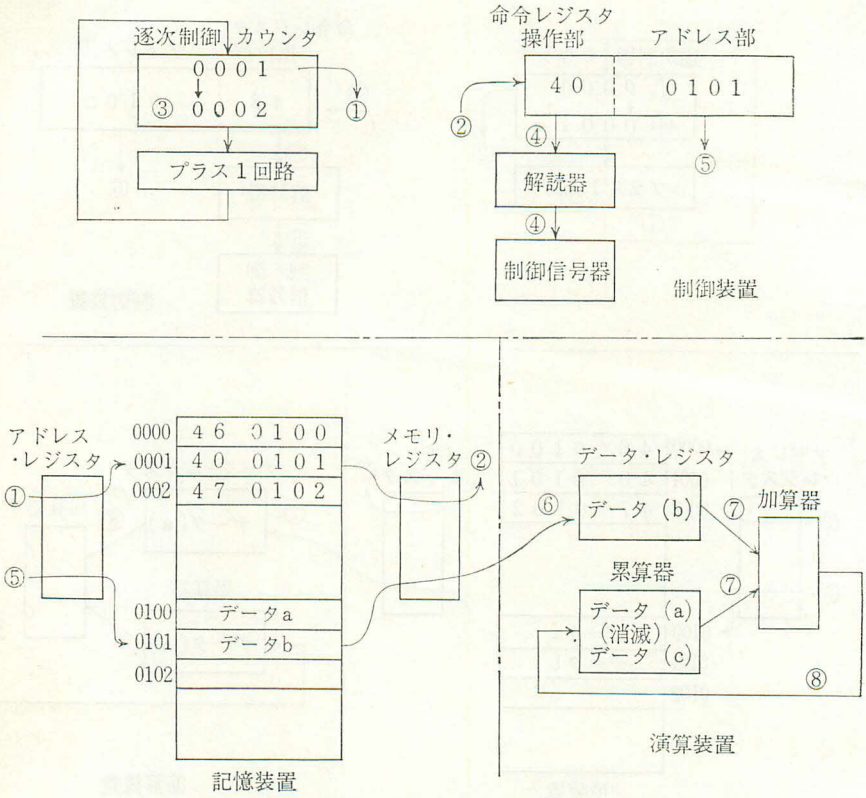
第2.93図 $(0000)_8$ 番地の命令の読み出しと実行

れ，同時に記憶装置に対して読み出し指令が出される。

⑦ 指定されたアドレスの内容データ (a) がメモリ・レジスタに読み出され，データ・レジスタへ送られる。

⑧ 別に，累算器には 0 がはいり（累算器は，最初にご破算される），この累算器とデータ・レジスタの内容が同時に加算器へ送られ加算される。加算の結果は，累算器にはいる。すなわち，累算器にはデータ (a) がはいる。

4. $(0001)_8$ 番地の命令の読み出しと実行 $(0001)_8$ 番地に記憶されている「$(0101)_8$ 番地のデータを加えよ」の命令は，**第2.94図**に示すように読み出され，実行される。すなわち，



第2.94図　$(0001)_8$ 番地の命令の読み出しと実行

① 逐次制御カウンタの内容は $(0001)_8$ となっており，これをアドレス・レジスタへ送り，同時に読み出し指令を出す。

② $(0001)_8$ 番地の命令がメモリ・レジスタに読み出され，命令レジスタへはいる。

③ 逐次制御カウンタの内容に1が加えられ $(0002)_8$ となる。

④ 命令レジスタの上位6ビット $(40)_8$ が解読器へ送られ，解読される。「加算」動作であることがわかるので，制御信号器から加算に必要な制御信号が出る。

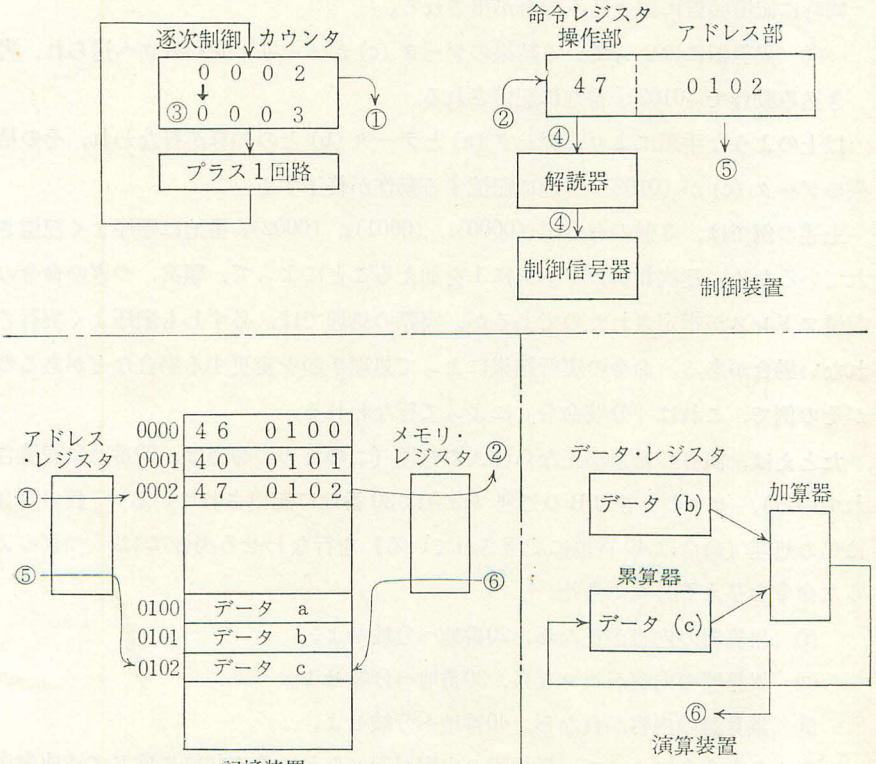
⑤ 命令レジスタの下位12ビット $(0101)_8$ がアドレス・レジスタへ送られ，同時に記憶装置に読み出し指令が出される。

⑥ 指定されたアドレスの内容データ (b) が，メモリ・レジスタに読み出され，データ・レジスタへ送られる。

⑦ 累算器にはいっているデータ (a) と，データ・レジスタにあるデータ (b) とが同時に加算器へ送られ加算される。

⑧ 結果のデータ (c) は，累算器にはいる。前に累算器に記憶されていたデータ (a) は消滅する。

5. $(0002)_8$ 番地の命令の読み出しと実行 $(0002)_8$ 番地に記憶されている「$(0102)_8$ 番地へ結果を記憶せよ」の命令は，**第 2.95 図** に示すように読み出され，実行される。すなわち，



第 2.95 図 $(0002)_8$ 番地の命令の読み出しと実行

① 逐次制御カウンタの内容は $(0002)_8$ となっており，これをアドレス・レジスタへ送り，同時に読み出し指令を出す。

② $(0002)_8$ 番地の命令がメモリ・レジスタに読み出され，命令レジスタにはいる。

③ 逐次制御カウンタの内容に1が加えられ，$(0003)_8$ となる。

④ 命令レジスタの上位6ビット $(47)_8$ が解読器へ送られ，解読される。その結果，「記憶」動作を行なうべきことがわかるので，制御信号器は，その制御を行なうための制御信号を発生する。

⑤ 命令レジスタの下位12ビット $(0102)_8$ がアドレス・レジスタへ送られ，同時に記憶装置に書き込み指令が出される。

⑥ 累算器にはいっている結果のデータ (c) がメモリ・レジスタへ送られ，書き込み動作で $(0102)_8$ 番地に記憶される。

以上のような手順により，データ (a) とデータ (b) との加算が行なわれ，その結果のデータ (c) が $(0102)_8$ 番地に記憶する動作が終了する。

上述の例では，3個の命令が $(0000)_8$，$(0001)_8$，$(0002)_8$ 番地に順序よく記憶されているため，逐次制御カウンタに1を加えることによって，順次，つぎの命令の記憶アドレスが指定されたのであるが，実際の処理では，必ずしも順序よく実行されない場合がある。命令の実行結果によって処理手順を変更する場合などがあるのがその例で，これは「分岐命令」によって行なわれる。

たとえば，演算の結果が正ならばAの処理（このための命令は，20番地に記憶されている），ゼロのときはBの処理（命令は30番地に記憶されている），負の場合はCの処理（命令は40番地に記憶されている）を行なわせる場合には，つぎのような命令を与える必要がある。

① 累算器の内容が正なら，20番地へ分岐せよ。

② 累算器の内容がゼロなら，30番地へ分岐せよ。

③ 累算器の内容が負なら，40番地へ分岐せよ。

このような命令によって，累算器の内容が調べられ，その状態に応じて逐次制御カウンタへ20，30，あるいは40の数字を入れることにより，結果のいかんによっ

て処理手順を変更することができる。

これはとりもなおさず，電子計算機が判断を行なって，処理手順を変更していることにほかならない。電子計算機が判断を行なって，種々の高度な処理を行なえるのも，このような飛び越しまたは**分岐命令**(1)が使えることによるといえる。

2.6.7 周辺装置の制御

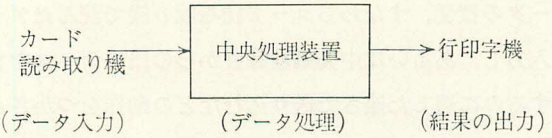
前項においては，演算を行なう場合の制御の方法を説明したが，本項では，周辺装置を制御する方法について説明する。

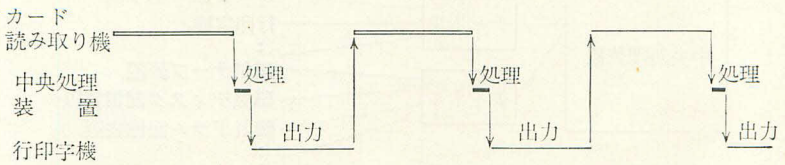
周辺装置を制御を行なう場合にも，プログラムの命令によって制御が行なわれることには変わりがなく，また，命令の読み出しおよび実行という2段階で行なわれることも演算の制御と同様であるが，特に周辺装置の制御という項を設けたのはつぎの理由による。

（1） 演算の処理は電子的な速度で行なわれるのに対し，周辺装置の動作は機械的な速度でしか行なうことができず，その結果，制御の方法も異なったものになる。

（2） 複数の周辺装置を同時に動作させる場合があり，制御が複雑になる。

いま，カード読み取り機でデータを読んで，それを処理し，結果を行印字機で印刷する場合を考えると，最も簡単な方法としては，**第2.96図**のように，入力し，



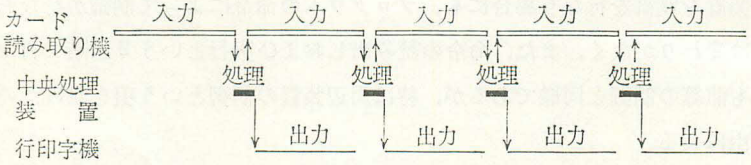


第2.96図　直列的な動作

(1) jump instruction

処理し，そして出力する動作のくり返しを行なうことが考えられるが，この場合には，中央処理装置と周辺装置が直列的に動作するため，中央処理装置は大部分の時間を遊んでいることになり，きわめて能率が悪い。

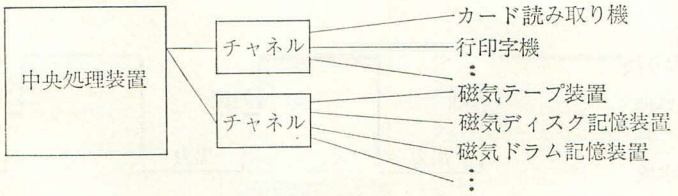
これに対し，**第2.97図**に示すように，入力したデータの処理を行なうのと同時にカード読み取り機を動作させ，処理が終了した場合には，カード読み取り機の動作と並行して行印字機を動作させるというように，中央処理装置と周辺装置とを並行的に動作させれば，各装置の使用効率は大幅に向上する。



第2.97図　並列的な処理

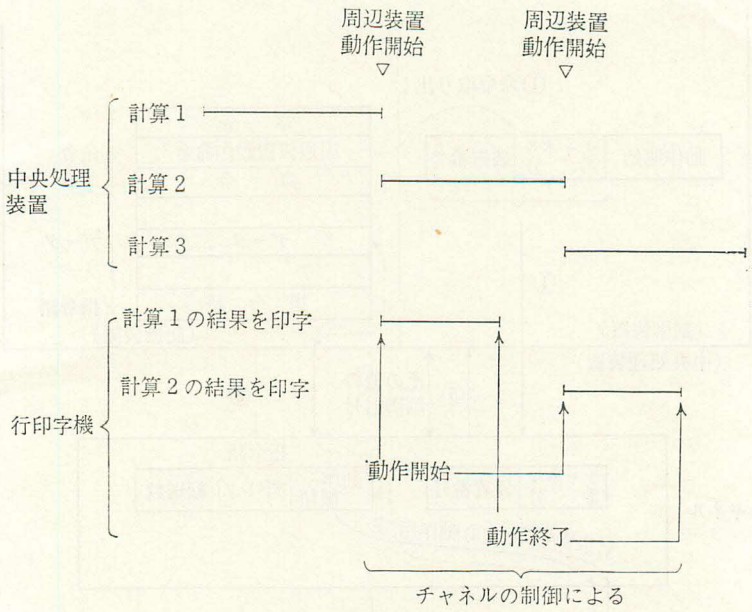
このような並行動作を可能にするため，電子計算機は**チャネル**(1)という装置によって，周辺装置の制御を行なっている。すなわち，**第2.98図**に示すように，中央処理装置と多数の周辺装置との間にチャネルと呼ばれる装置を設けておき，プログラムで周辺装置を動作させる命令が出された場合には，周辺装置の制御はチャネルにまかせてしまい，制御装置はデータの処理に専念する。一方，チャネルは周辺装置の動作を制御し，データの授受，すなわちコード読み取り機で読んだデータをまとめて中央処理装置へ入力し，あるいは中央処理装置からの出力データをまとめて受け，行印字機の印刷するのに適した速さで送り込むなどの動作をつかさどる。

このように制御の機能を分割することにより，周辺装置の動作と中央処理装置内



第2.98図　チ　ャ　ネ　ル

(1)　channel



第 2.99 図　行印字機動作の制御

での演算処理とを並列的に進行させることができる。たとえば**第 2.99 図**に示すように，中央処理装置がある計算を終了し，行印字機で印刷する必要が生じたとき，中央処理装置は周辺装置の動作開始を指令する。行印字機が接続されているチャネルがその指令を受けて行印字機を動作させる一方で，中央処理装置はつぎの処理に移ることになる。

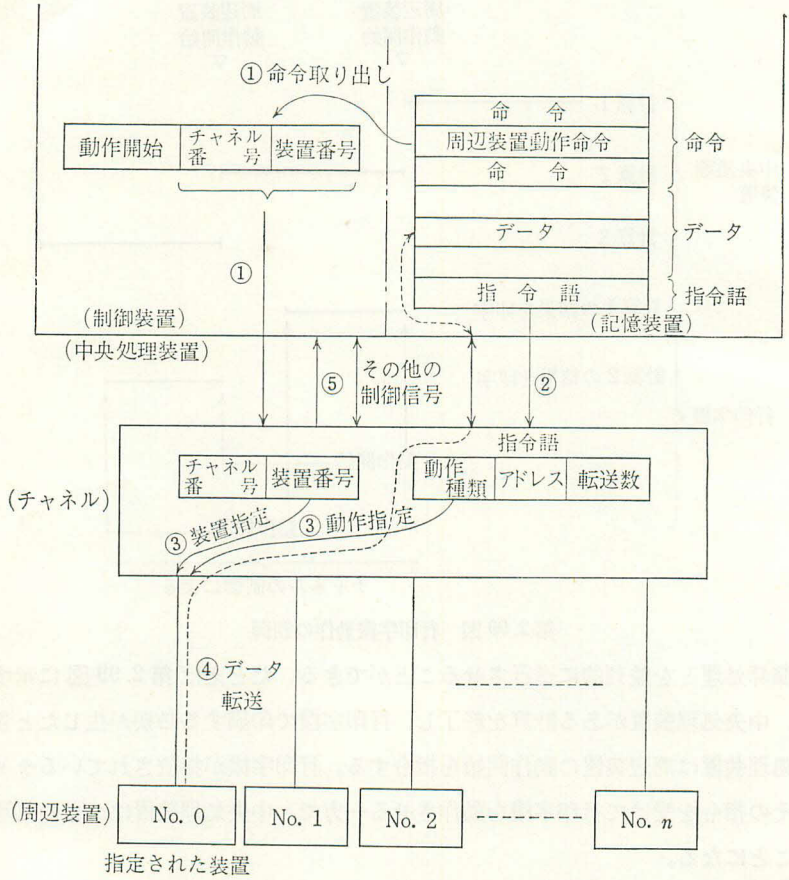
第 2.100 図は，周辺装置の制御方法を具体的に示したものである。

周辺装置を動作させるための命令は，

操作部	アドレス部	
動作開始	チャネル番号	装置番号

操作部：周辺装置の動作開始を指示する。

アドレス部：いくつか設備されているチャネルのうち，動作を開始させるべき装置が接続されているチャネルの番号と，その装置の番号とを指定する。



第2.100図　周辺装置の制御

で構成されており，命令はつぎのように実行される。

①　命令が読み出され，操作部の内容によって周辺装置を動作させる命令であることがわかると，指定のチャネルに対してアドレス部の内容を送り，チャネルはこれを記憶する。

②　チャネルは，主記憶装置にあらかじめ記憶されている指令語（コマンドともいう）を読み出し，それを記憶する。

指令は，たとえばつぎのような形式になっている。

動作の種類	記憶装置のアドレス	転送数

動作の種類：「読め」，「書け」など，周辺装置が行なうべき動作の種類を指定する。

記憶装置のアドレス：入力データを記憶し，あるいは出力データが記憶されている記憶装置のアドレスを指定する。

転送数：入力装置から入力する字数，あるいは出力装置へ出力する字数を指定する。

ここで，中央処理装置とチャネルとの連絡は断たれ，中央処理装置は別の処理にかかり，その後の周辺装置の制御はチャネルが行なう。

③　チャネルは，指定された周辺装置に対して，指令語で指定されている動作を指示し，それを受けて周辺装置は動作を開始する。

④　入力の場合には，周辺装置で読まれたデータがチャネルを経由して，指令語によって指定されている記憶装置のアドレスに記憶される。

出力の場合には，記憶装置に記憶されているデータがチャネルを経由して，周辺装置へ送られる。

チャネルは，転送のたびにその字数を数え，指令語で指定されている転送数と比較して，両者が等しくなれば，データの転送が終了したものと判断する。

⑤　転送が終わると，チャネルは中央処理装置に対して終了したことを知らせる信号を送る。

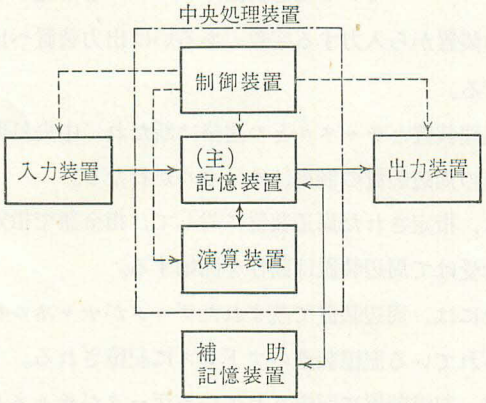
以上のようなチャネル制御により，周辺装置の動作，データの転送が行なわれる。

2.7 摘　　要

1. 電子計算機でデータの処理を行なうには，その処理手順を前もってプログラムとして電子計算機に覚え込ませる必要がある。このように，あらかじめプログラムを記憶しておく方式の計算機を，プログラム記憶式計算機という。
2. 電子計算機で用いるデータは，すべて2進数として表現される。
3. 電子計算機は，ハードウェアと，ソフトウェアの2つの体系で構成されている。

このうち，ハードウェアは，つぎの各装置で構成されている。

- 電子計算機のハードウェア
  - 中央処理装置
    - 制御装置
    - (主)記憶装置
    - 演算装置
  - 周辺装置
    - 補助記憶装置
    - 入出力装置
      - 入力装置
      - 出力装置



第2.101図

これら各装置の機能は，つぎのようである。

（1）　入力装置：紙カードや紙テープに記録されているデータを読んで，それを処理につごうのよい電気信号に変換したうえで，中央処理装置へ送り込む装置である。

（2）　記憶装置：記憶装置は，主記憶装置と補助記憶装置とに大別され，前者は中央処理装置の中にあって，当面の処理に必要なプログラムやデータを記憶する装置であり，後者は中央処理装置の外にあって，さしあたって必要がなく，主記憶装置に記憶しきれない大量のデータを記憶する装置である。

（3）　演算装置：演算装置は，中央処理装置の中にあって，四則演算や，その他の演算を行なう装置である。

（4）　制御装置：制御装置は，電子計算機の各装置の動作を制御するための装置である。プログラム記憶式計算機では，処理手順（プログラム）を記憶しておき，それにしたがって制御装置が必要な制御を行なう。

（5）　出力装置：中央処理装置で処理された結果は，電気信号となって出力装置へ送られ適当な符号に変換されたうえで，用紙に印刷されたり，紙カードにせん孔される。

4.　電子計算機の主記憶装置には，磁心を記憶素子とする磁心記憶装置が広く用いられている。そして，磁心の磁化方向（磁化が時計方向か，反時計方向か）によって，記憶が行な

われる。磁心1個では，1ビットのデータ（“1”か，“0”か）を記憶することができる。

5. データを磁心に記憶することを書き込みといい，逆に，磁心に記憶されているデータを取り出すことを読み出しという。これらの動作は，磁心の中を通る4本の導線のうち，X，YおよびZ線の電流を制御することで行なわれる。記憶内容は，S線に読み出される。

6. 磁心に記憶されている内容を読み出せば，その内容は破壊される。このような読み出しの方法を，破壊読み出しという。したがって，読み出したあとは，再書き込みにより記憶内容を復元する必要がある。

7. 磁心は，1枚の板に，たとえば縦横64個ずつ（64×64=4 096）とか，128個ずつ(128×128=16 384）というように，整然と並べられ，コア・プレーンを形成している。

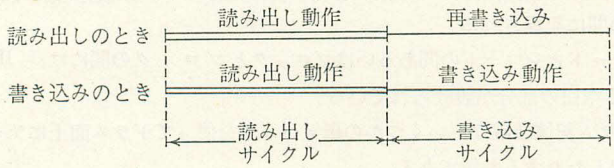
このような多数の磁心の中から，読み書きの対象となる特定の磁心を選び出すには，該当する磁心のX線，Y線に電流を流す電流一致方式によって行なう。

8. コア・プレーンを必要な枚数だけ積み重ねたものをスタックという。スタックを構成するのに，何枚のコア・プレーンを積み重ねるかは，その計算機でどのようなデータを取り扱うかによって決まる。たとえば，1字6ビットのデータを取り扱うには，6枚のコア・プレーンを積み重ねることになる。

9. 磁心記憶装置には，番地がつけられている。7. で例に上げたコア・プレーンを，スタックに組み立てた場合，前者では0～4 095番地，また，後者では，0～16 383番地まで，番地づけされる。

データの書き込み，読み出しは，プログラムで番地を指定して行なわれる。

10. 磁心記憶装置の動作は，読み出しおよび書き込みの2つのサイクルに分けて行なわれる。



第2.102図

11. 記憶装置に記憶されている内容を読み出しても，再書き込みの動作でその内容を復元するので，内容は変わることはない。しかし，新しいデータを書き込めば，その番地に記憶されていた古い内容は失われて，新しい内容に変わる。

12. 記憶装置に対して読み出しを指示してから，記憶装置で読み出し動作と再書き込みが行なわれ，つづいて，つぎの読み出しのための準備が完了して読み出し指示が受けつけ可能となるまでの時間をサイクル・タイムといい，普通1～数 μs であり，最近では 1μs 以下のものも使われる。

また，読み出しを指示してから，要求のあった記憶内容が取り出されるまでの時間をアク

セス・タイムといい，サイクル・タイムのほぼ半分程度である。

13. 磁心記憶装置に記憶されるデータは，パリティ・チェックによって，1字ごと，または1バイト，あるいは何字かの組ごとに誤りの有無が検査される。記憶装置は，パリティ・ビット用のコア・プレーンを持っている。

14. 主記憶装置の容量を補なうための補助記憶装置として，現在広く用いられているものには，磁気テープ装置，磁気ドラム記憶装置，磁気ディスク記憶装置などがある。これらの装置は，いずれも，磁気ヘッドを用いて磁気記録媒体上にデータの書き込み，読み出しを行なう。

15. 磁気記録媒体上のデータの記録方式には，NRZ方式，RZ方式あるいは位相変調方式などがある。

16. 磁気テープ装置は，磁気記録媒体として磁気テープを使用し，データの読み出しあるいは書き込みを行なう装置である。

磁気テープは装置に自由に着脱できるので，何巻かの磁気テープを使うことによって，きわめて大量のデータを記憶することができ，また，装置のビット当たりの価格がきわめて安いことも大きな利点である。

しかし，ほしいデータを読み出すまでに長時間を必要とするのが欠点である。磁気テープ装置のように，始めから順にデータの読み書きをする装置をシーケンシャル・アクセス装置という。

17. 磁気テープ装置は，7あるいは9トラック用の磁気ヘッドを用いて，データの読み書きを行なう。データが記録される道をトラックといい，1字あるいは1バイトのデータは，磁気テープの幅方向（列）に記録される。

また，磁気テープの始端には BOT が，また，終端には EOT が設けられており，データは，この両者の間に記録される。

さらに，レコードとレコードの間あるいはブロックとブロックの間には，IRG あるいは IBG といわれる空白の部分が設けられている。

18. 磁気ドラム記憶装置は，いくつかの磁気ヘッドを使ってドラム面上にデータを書き込んだり，読み出したりする装置である。

磁気ドラム記憶装置は，データを書き込むべき場所，あるいは読み出したいデータが記憶されている場所をすばやく見つけることができるのが特徴で，このような特徴を持つ装置をランダム・アクセス記憶装置と呼んでいる。

19. データはドラム円周上のトラックに記録されるが，その形式には，ビット直列，ビット並列および直並行形式がある。

20. 磁気ディスク記憶装置は，磁性記録媒体として何枚かの円板（ディスク）を重ねたものを用い，アームに取りつけられた磁気ヘッドによってディスク表面にデータの書き込み，読み出しを行なう装置である。ディスクを着脱できるものを特にディスク・パックという。

磁気ドラム記憶装置は，読み書きに要する時間は最も短いが，記憶容量に限界があり，ま

た，磁気テープ装置は，記憶容量は大きいがデータの読み書きに時間を要したが，磁気ディスク記憶装置は，記憶容量が大きくしかも読み書きに要する時間が短いのが特徴である。

21. データは，仮想円筒上（シリンダ）のトラックに記録される。このシリンダは，アクセス・アームの移動（磁気ヘッドの移動）によって指定される。

22. カード読み取り機は，紙カードにせん孔されているデータを読んで，中央処理装置へ入力するための装置である。

カードせん孔機は，中央処理装置で処理された結果を紙カードにせん孔するための装置である。

23. テープ読み取り機は，紙テープにせん孔されているデータを読んで，中央処理装置へ入力するのに用いられる。

テープせん孔機は，中央処理装置で処理された結果を，紙テープにせん孔するのに用いられる。

24. 行印字機は，中央処理装置で処理された結果を用紙に印刷するための装置である。

25. 制御卓は，電子計算機の動作状態を監視したり，操作員が必要に応じて動作に介入するために設けられた装置である。

26. 演算装置は，四則演算や，論理演算を行なうための装置である。基本的な考え方は卓上計算機の演算回路と同じものであるが，演算速度を上げるための種々のくふうがなされている。たとえば，並列加算方式や，けた上げ先取り方式などはその例である。

27. 制御装置は，電子計算機の各装置の制御を行なう装置である。この制御は，人間が作ってあらかじめ電子計算機に与えてあるプログラムに基づいて行なわれる。

28. プログラムは，プログラム言語を使って作られる。これには，機械語，アセンブリ語およびコンパイラ語があるが，電子計算機は，このうち機械語のプログラムしか命令として受けつけてくれない。したがって，アセンブリ語やコンパイラ語のプログラムは，計算機によって機械語のプログラムに翻訳する。

29. 機械語の命令は，操作部とアドレス部とで構成されており，前者は，計算機に行なわせる動作の種類を指示し，後者は，処理の対象となるデータが記憶されている記憶装置のアドレスを指定する。

これらは，いずれも2進数として表示されている。

30. プログラム記憶式計算機では，データの処理に先だって，機械語の命令を記憶装置に記憶する必要がある。これをプログラムのロードという。この場合，命令は，プログラミングした順序どおりに記憶される。

データ処理を行なう場合には，この命令群を最初のものから順番に取り出し実行していけば，われわれが考えた手順どおりに処理が進行することになる。

31. データの処理を行なう場合には，記憶装置に記憶されている命令を取り出す命令取り出し段階と，その命令によりデータの処理に必要な制御を行なう命令実行段階の2つの段階がある。

32. 命令の取り出しは，逐次制御カウンタによって制御される。すなわち，最初の命令が記憶されているアドレスを逐次制御カウンタで指定し，その後は，逐次制御カウンタの内容を自動的に増加していくことによって，順次命令が取り出され，実行されることになる。このように，順次命令を取り出して制御を行なって行く方式を逐次制御という。なお，処理順序を変更するには，分岐命令によって，逐次制御カウンタの内容を変更すればよい。

33. 中央処理装置での演算処理と，周辺装置の動作を並行的に行なわせ，計算機の使用効率を向上するために，チャンネルを設ける。

2.8 練習問題

1. 主記憶装置のアクセス・タイムと，サイクル・タイムについて説明せよ。

2. ある電子計算機のカタログに，記憶容量約16kバイト（1バイト8ビット，パリティ1ビット）と書かれていた。

この記憶装置について，次表に適当な数字を入れよ。

プレーン内の磁心配列	128×128
X線の本数	(1)　本
Y線の本数	(2)　本
Z線の本数	(3)　本
S線の本数	(4)　本
磁心の数	(5)　個
アドレス	(6)　番地～　番地
記憶容量	(7)　バイト

3. つぎの問に答えよ。

(1) 1分間5000回転の磁気ドラム記憶装置のアクセス・タイムは，何 ms か。

(2) 1分間3000回転，アクセス・アームの移動に要する時間平均 100ms の磁気ディスク記憶装置のアクセス・タイムは，何 ms か。

(3) リール1巻700mの磁気テープの終端のデータを読むまでに何秒かかるか。ただし，テープは平均毎秒3mで走行するものとする。

4. つぎの各装置のうち，ランダムにデータの読み書きができるものに○印をつけよ。

(1) カード読み取り機

(2) 磁気ディスク記憶装置

(3) 行印字機

(4) 磁気ドラム記憶装置

（5） テープ読み取り機
（6） カードせん孔機
（7） 磁気テープ装置
（8） テープせん孔機

5. 機械語の命令は，操作部とアドレス部で構成されているが，それぞれの役目は何か。

6. つぎの□の中に，下に列記したなかから適当な語を選んで入れよ。

プログラム記憶式計算機では，プログラムの各命令は，(1) に記憶されている。その実行に際しては，まず (2) 段階で命令が (3) の (4) に取り出され，続いて，(5) 段階で，命令の (6) が (7) によって解読され，その命令実行に必要な制御が行なわれる。

A. 制御装置　B. 記憶装置　C. 演算装置
D. 命令実行　E. 解読器　F. 加算器
G. 命令取り出し　H. 操作部　I. アキュムレータ
J. アドレス部　K. 命令レジスタ　L. 逐次制御カウンタ

7. 逐次制御と分岐について，簡単に説明せよ。

第3章　高性能の電子計算機

第2章においては，一般的な電子計算機の基本構成と動作について説明した。ところで現在では，情報処理システムの複雑化にともない，これらの基本的な機能の上にさらに複雑な機能を持った電子計算機が要求されるようになった。そこで第3章では，このような高性能の電子計算機について考えてみよう。まずはじめに，電子計算機を利用するシステムがどんなものであるかについて，最近特に重要になってきている**オンライン・リアルタイム・システム**(1)を中心に説明し，ついで高性能の電子計算機が持っていなければならない機能について述べる。そして最後に，オンライン・リアルタイム・システムにとって欠くことのできない**データ伝送**(2)と**端末装置**(3)について触れることにする。

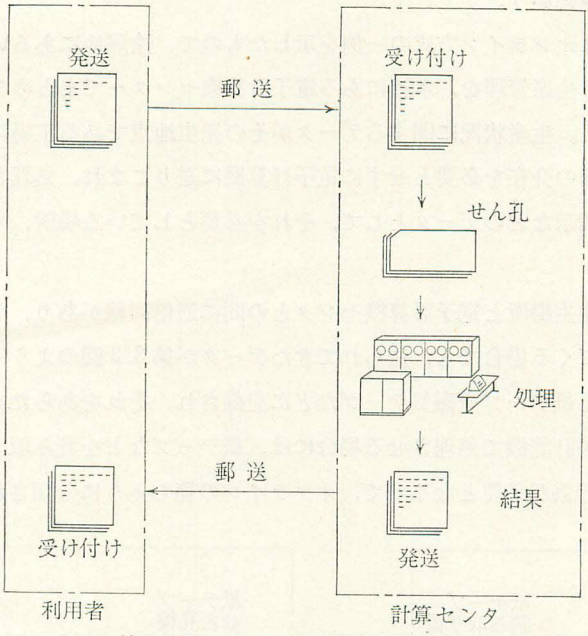
3.1　電子計算機を利用するシステム

3.1.1　オンラインとリアルタイム

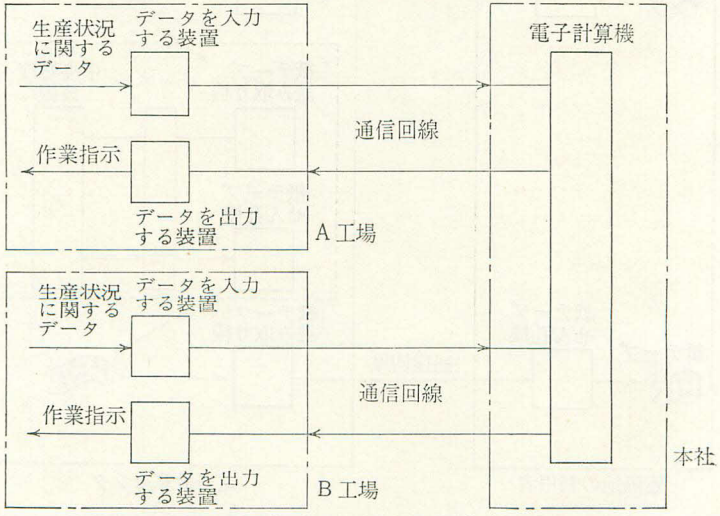
1.　オンラインとオフライン(4)　第2章で説明した電子計算機によりデータを処理する場合には，第3.1図に示すようにデータを電子計算機のあるセンタまで郵送したり，人間が運んだりしなければならない。このように，遠隔地で発生したデータを電子計算機に投入する過程において，あるいは電子計算機で処理した結果を遠隔地に送出する過程において，人手の介在を必要とする場合を**オフライン**という。

しかるに，電子計算機の利用範囲が拡大するにつれて上記のセンタとデータが発生する場所，あるいはセンタと処理した結果を必要とする場所とが，離れている場合が一般的になってきた。さらに，データをなるべく早く処理し，その結果をすぐに手に入れたいという要望も生じてきた。こうなると，第3.1図のオフライン方式ではデータを伝送するための時間がかかり，また，人手も要するので，データをその発生地点から直接電子計算機に投入しようという思想が生れた。このような方式

(1)　on line real time system　(2)　data transmision　(3)　terminal unit
(4)　off line



第3.1図 オフライン処理の例（1）

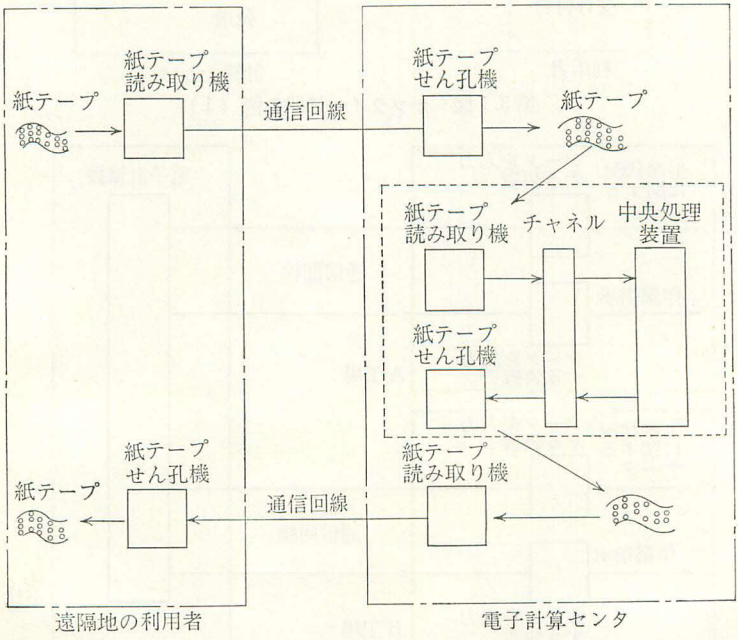


第3.2図 オンライン処理の例

をオンラインという。

第3.2図はオンライン方式の一例を示したもので，遠隔地にあるいくつかの製造工場における生産管理を，本社にある電子計算機センターでまとめて行なおうというものである。生産状況に関するデータがその発生地点である工場において入力され，途中人間の介在を必要とせずに電子計算機に送りこまれ，処理された後，その結果が作業指示などのデータとして，それを必要としている場所，すなわち工場に送られる。

データの発生場所と電子計算機センタとの間に通信回線があり，データが電気的に伝送されてくる場合でも，送られてきたデータが第3.3図のようにセンタにおいて，いったん紙テープや磁気テープなどに記録され，それをあらためて読み取り機にかけて電子計算機で処理させる場合には，紙テープなどを読み取り機にかけるという人間の行為が必要となるので，オフラインの範ちゅうに分類される。また，記



第3.3図　オフライン処理の例　(2)

録された紙テープなどをただちにかけ替えて処理する場合を，とくに区別してインライン処理と呼ぶことがある。

2. リアルタイムとディレィドタイム(1)　第3.1図に示したオフライン処理の場合には，データを郵送してから答が返送されてくるまで数日を要することになるが，月ごとに作成する統計資料のように緊急を要しないものについては，これでもとくに不便は感じないであろう。しかし，列車の指定券を電子計算機を使って発売するような場合には，窓口に並んでいるお客を長い間待たせておくわけにはいかないから，第3.1図のような方法をとることは許されない。

そこで，窓口とセンタにある電子計算機を通信回線で結び，指定券を発売するのに必要なデータ（たとえば，乗車列車名，月日，区間など）を窓口から送り込めば，ただちにセンタの電子計算機がこれを処理して，お客に指定券の有無を答えられるようにしておかなければならない。

このようにデータを送り込むと，即座にその結果が送り返される場合，これを**リアルタイム処理（実時間処理）**(2)という。これに対して，答が出るまでにいくばくかの時間を要する場合，これを**ディレィドタイム処理**(3)という。

これまで「即座に」という言葉を用いたが，同じリアルタイムといっても答がもどってくるまでの時間は一義的に定義されていない。たとえば，上にのべた指定券を発売する場合には数秒以内に解答が返ってくればよいが，電子計算機を利用してロケットの追跡を行なうような場合には，結果が得られるまでの時間を数ミリ秒以内におさえる必要がある。すなわち，電子計算機を利用する業務が秒単位の仕事であるか，ミリ単位の仕事であるかによって，「即座に」の内容が変わってくるのである。

以上の点を含んでリアルタイム処理という言葉を定義しなおすと，「リアルタイム処理とは，外部の事象の発生（たとえば，お客が窓口に並んだり，ロケットが発射されるなど）に応じて，要求される時間内にデータを処理する方式である」ということができる。このようにリアルタイムという言葉は絶対的なものではなく，ディレィドタイムとの境界線はそれほど明確ではない。

なお，リアルタイム処理を行なうにはオンラインか，あるいはインラインである

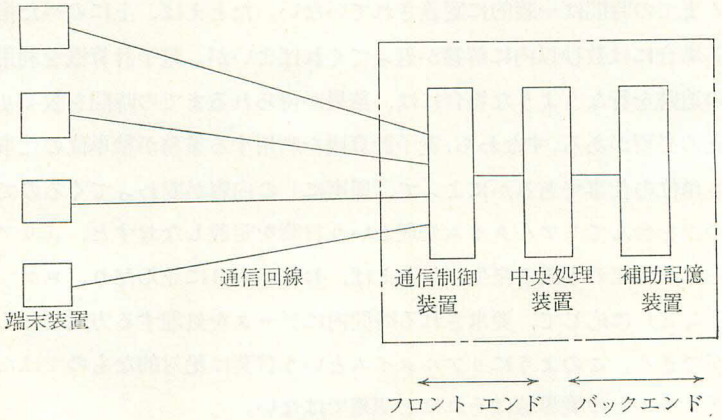
（1） delayed time　（2） real time processing　（3） delayed time processing

必要があるが，逆にオンラインであるからといって必ずしもリアルタイムであるとは限らない。オンラインではあるがリアルタイムでない場合については，**3.1.2**の一括処理の項において述べることにする。

3.　オンライン・リアルタイム・システムとその構成　　第3.2図に示したオンライン処理の例では，電子計算機のほかに，各工場と本社を結ぶ通信回線，および工場においてデータを入出力する装置が必要であった。これら3種の装置は，どれが欠けてもオンライン処理を行なうことはできず，3者一体となってはじめてオンライン処理が可能となる。

このように，各構成要素が有機的に結びつけられ，全体としてある目的に沿った働きをするものを**システム**と呼んでいる。システムという言葉を最近いろいろな場合に用いられるようになってきたが，電子計算機に関するものでは，たとえば電子計算機によって事務処理を行なう **EDPS（電子データ処理システム）**[(1)] などの例がある。同様に，オンライン処理を行なうための各種の装置の総体を**オンライン・システム**といい，特にリアルタイム処理を行なう場合を**オンライン・リアルタイム・システム**といっている。

第3.4図にオンライン・システムの代表的構成を示す。同図の端末装置は，第3.2



第3.4図　オンライン・システム

(1)　electronic data processing system：略して EDPS

図では各工場においてデータを入力したり出力したりする装置に相当し，データの発生する場所または情報を必要とする場所に設置される。これらの端末装置には各種の形式があるが，その詳細については3.4で述べる。

端末装置から入力されたデータは通信回線を通して電子計算機に送られ，また，電子計算機で処理された結果も通信回線を使って端末装置に送られるが，このような端末装置と電子計算機との間のデータの送受をデータ伝送といい，これについては3.3で説明する。

オンライン用の電子計算機には，オフライン用にはなかった**通信制御装置**(1)が必要である。この装置は，端末装置と**中央処理装置**(2)との間で，通信回線を通してデータのやりとりが円滑に行なわれるようにする役目を持っている。

オンライン・システムでは，データを入出力するという機能は周辺装置としての入出力装置よりは端末装置に移り，一方，補助記憶装置，特に磁気ディスク装置や磁気ドラム装置等の**ランダム・アクセス記憶装置**(3)が，システムの動作のなかで大きな位置を占めるようになる。なお，中央処理装置を中心にして考えたとき，通信制御装置側を**フロントエンド**(4)，補助記憶装置側を**バックエンド**(5)という。

オンライン・システムは，電子計算機に関する技術すなわち情報処理技術と，通信回線を通して情報を送り受けする技術すなわち通信技術の2つが結びついてできたものである。このように情報処理と通信という2つの分野が融合したシステムに対して，**データ通信**(6)という言葉が使われるようになった。従来，通信といえば，電話のように人間と人間との間の通信に限られていたが，データ通信では機械と人間の間，さらには機械と機械との間の通信にまで対象が広げられている点に特徴がある。

3.1.2 一括処理（バッチ処理）システム(7)

高価な電子計算機を使用する際には，電子計算機が効率的に運用され，遊ぶ時間が多くならないよう，計画的に仕事を行なわせることが必要である。それには，処理すべきデータが全部集ってから，それをまとめて順序よく電子計算機に処理を行な

---

(1) communication control unit (2) central processing unit
(3) randum access memory (4) front end (5) back end
(6) data communication (7) batch processing

わせるのが望ましい。このような処理の方法を**一括処理**または**バッチ処理**という。

仕事を一括して行なう性質上，この処理はリアルタイム処理とはならない。しかし，データの入出力は，オフラインであってもオンラインであってもよい。

第3.5図(a)の例はオフラインによるバッチ処理の例で，このシステムの利用者は処理すべきデータを電子計算機のあるセンタに持っていったり郵送したりする。

図(b)に示す例は，オンライン・リアルタイム処理においてバッチ処理が併用されているもので，1か月分あるいは1年分のデータをまとめた統計表の作成等は，オンライン・リアルタイム処理のあいまに生ずる電子計算機の余裕を利用してバッチ処理が行なわれる。なおこのように，主となるオンライン・リアルタイム処理のあいまに行なわれるバッチ処理を，**バックグランド処理**(1)と呼んでいる。

図(c)に示す例はオンラインによるバッチ処理の例で，**リモートバッチ処理**といわれている。情報が発生する場所から通信回線を通して伝送されてきたデータは，補助記憶装置に記憶され，順番がくると処理が行なわれる。結果は再び通信回線を通して伝送され，行印字機等による印字が行なわれる。情報を投入しまた受け取る場所においては多量のデータの送受が行なわれるので，一般の電子計算機で使用されている入出力装置と同様な装置または小形の電子計算機が使用される。

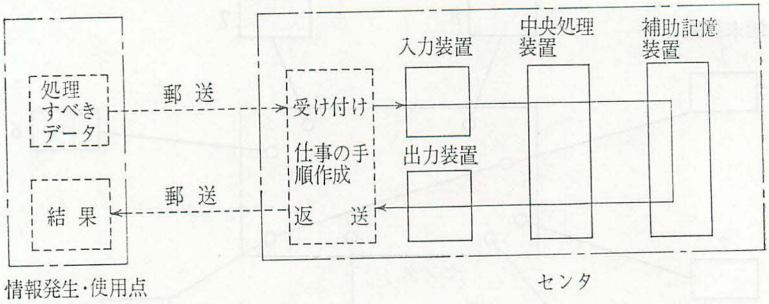
リモートバッチ処理では，せまい意味でのデータだけを扱うシステムと，プログラムも投入できるシステムとがある。

3.1.3　メッセージの交換を行なうシステム

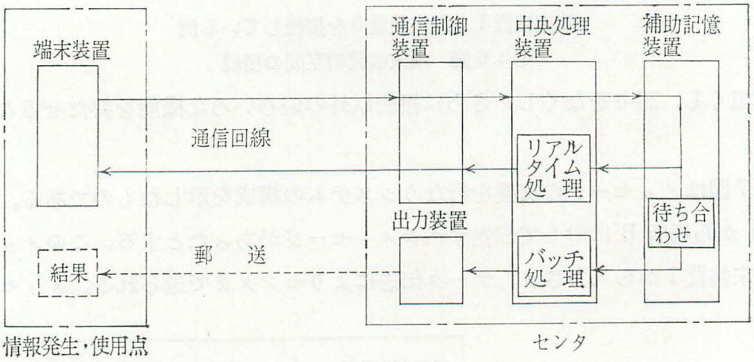
たとえば，電報の1通のようにある意味を表わすデータのひとかたまりを**メッセージ**(2)という。このようなメッセージを遠隔地へ電気的に送るには，端末装置と通信回線が必要である。端末装置が2つだけの場合は，両者を結ぶ1本の通信回線を設ければよいが，端末装置が多数になると各端末間相互をそれぞれ結んだのではきわめて多くの通信回線が必要となるので，第3.6図に示すように通信回線を1個所に集め，通信の相手を変更するための接続換えを行なう交換装置が必要となる。

接続換えを行なうだけなら通常の電話交換機をそのまま利用できるが，これを電子計算機を利用してオンライン・リアルタイム処理で行なわせることにより，接続

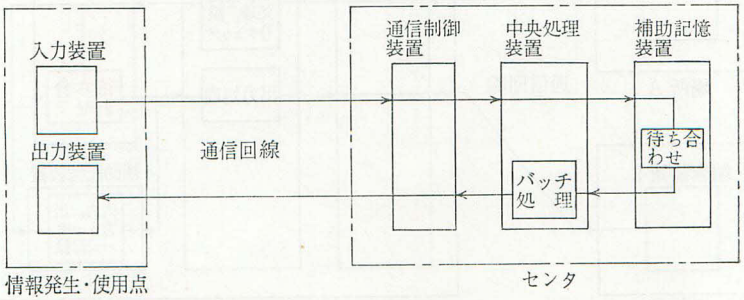
(1)　back ground processing　　(2)　message



(a) オフラインによるバッチ処理の例

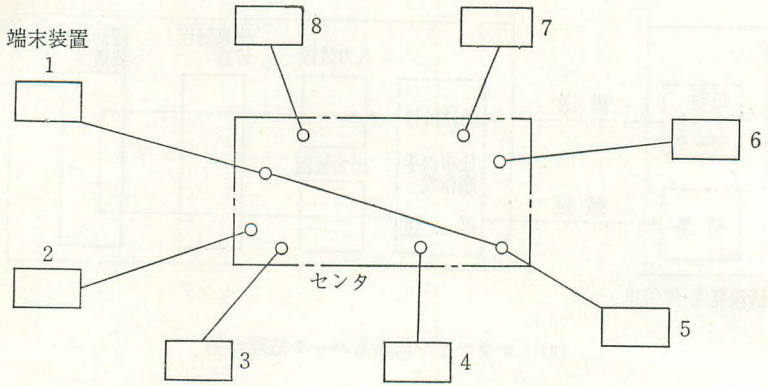


(b) オンライン・リアルタイム処理において併用されたバッチ処理の例



(c) リモートバッチ処理の例

第3.5図 バッチ処理システムの例

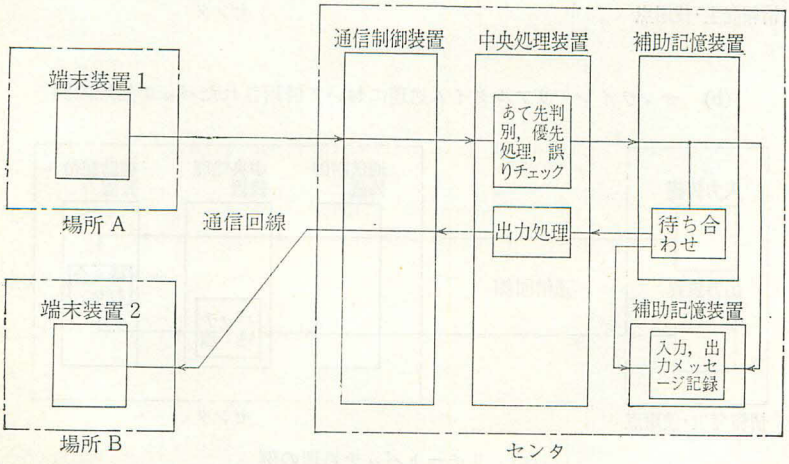


端末装置1と端末装置5を接続している例

第3.6図　端末装置相互間の接続

時間を短くし，誤りをなくし，さらに接続以外のいろいろな機能を持たせることができる。

第3.7図はメッセージの交換を行なうシステムの構成を示したものである。いま場所 A から場所 B に対して伝送すべきメッセージがあったとする。このメッセージは端末装置1から入力され，データ伝送によりセンタまで送られる。メッセージ



第3.7図　メッセージ交換システムの例

には本文のほか，宛先(この場合，端末装置2)，そのメッセージの優先度（たとえば至急扱いにするか普通でよいか等）などが含まれている。

まず，メッセージの内容に誤りがあるかないかが通信制御装置と中央処理装置で判定される。地名や会社の名前などで示された宛先は，中央処理装置で解読され，端末装置2であることがわかる。つぎに，優先順位が判定され順位の高いものから順番に相手の端末装置に送り出される。この際，相手がお話し中のときは，相手があくまでメッセージを補助記憶装置(普通はランダム・アクセス記憶装置)にたくわえておく。宛先が複数の場合，あるいは全端末装置の場合でも，また，接続すべき両方の端末装置のデータの送り方が異なっていたり速度に違いがあっても支障はない。

万一，障害が発生した場合に，もとの状態を再現するため，あるいは各種の統計を取るために，交換処理が行なわれるつど各種の記録が取られる。この記録は補助記憶装置（磁気テープ装置等）に記憶される。

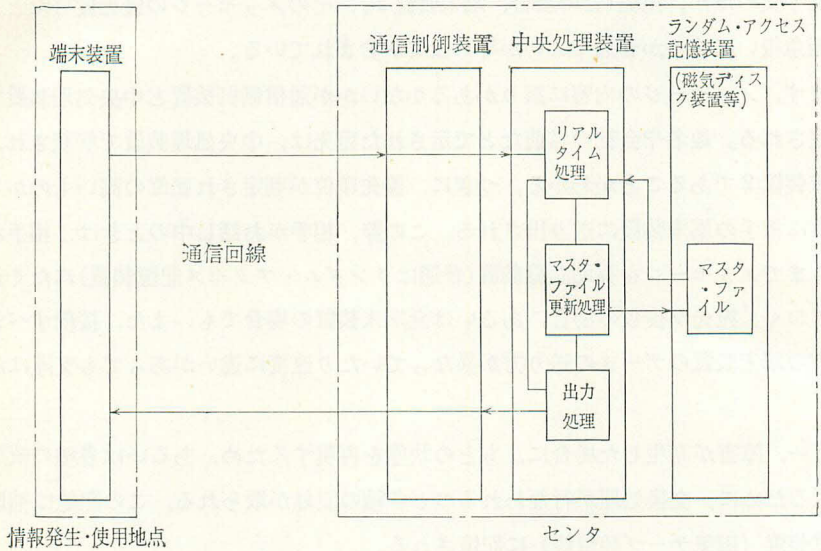
3.1.4 データの内容をリアルタイム処理するシステム

このシステムは照会システム，テレプロセッシング・システム等といわれるもので，その代表的な例として座席指定券の予約システムがある。

窓口にある端末装置から乗車月日，列車名，区間，等級などを指定して空席の有無について照会がなされると，センタの電子計算機は，ランダム・アクセス記憶装置に記憶されている該当列車の座席の台帳を調べ，空席が見つかると，その旨端末装置へ解答する。指定券を販売する場合は，端末装置で指定券の印刷が行なわれるとともに，その席に対応する補助記憶装置の記憶内容が空席から発行済に変更される。

第3.8図はリアルタイム処理システムの構成を示したものである。先の例で，該当列車の座席の台帳に相当するデータの集りのことを **マスタ・ファイル**(1) といい，第3.8図に示すように常時は磁気ディスク装置等のランダム・アクセス記憶装置に記憶されている。ランダム・アクセス記憶装置が使用される理由は，リアルタイム処理ではデータを読み出したり書き込む時間の短いことが要求されるからである。

(1) master file



第3.8図　リアルタイム処理システムの例

端末装置から送られてきたデータが，通信制御装置を経て中央処理装置にはいり主記憶装置に記憶されると，マスタ・ファイルの内容が主記憶装置に読み出され，その内容によって出力すべき結果が求められる。これは，先の例では空席ありの解答に相当する。一方，必要あらばマスタ・ファイルの内容が修正され，もとのランダム・アクセス記憶装置にもどされる。これは該当する座席に発売済みの印をつけることに相当している。

同じオンライン・リアルタイム・システムであっても，メッセージの交換を行なうシステムがデータの内容に関する処理をほとんど行なわないのに対し，このシステムではデータの内容についての処理が主体であり，マスタ・ファイルが重要な役割を果している。この種のシステムは予約業務のほか，オンライン預金業務，各種の情報提供業務用として使用される。

3.1.5　タイムシェアリング（時分割）・システム(1)

電子計算機を使用する場合，その能力と価格が常に問題にされる。電子計算機の

(1)　time-sharing system

能力（これを処理能力という）を評価することはそう簡単ではないが，たとえば適当な問題を解かせてその所要時間により判定したり，その電子計算機の各種の性能を分析して評価したりする。処理能力は高いほどよいが，価格が高くなるのが問題となるため，**価格と能力の比（コスト・パフォーマンス）**(1) が重要なポイントとなっている。

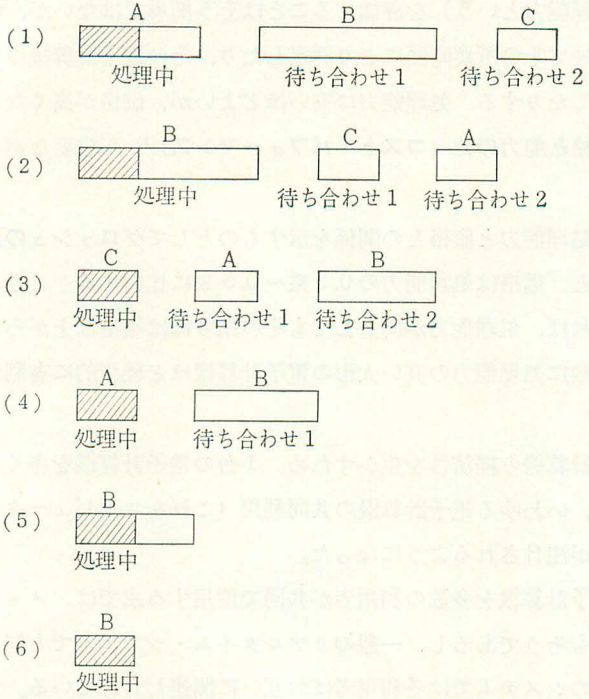
電子計算機の処理能力と価格との関係を示すものとして**グロッシュの法則**(2)がある。それによると「価格は処理能力の0.3乗～0.5乗に比例する」とされている。これはいいかえれば，処理能力が向上してもその割りには価格は上がらないということであり，一般に処理能力の高い大形の電子計算機ほど経済的に有利だということになる。

この大形電子計算機の経済性を生かすため，1台の電子計算機を多くの利用者が共同で利用する，いわゆる電子計算機の共同利用（これをコンピュータ・ユーティリティという）が注目されるようになった。

もっとも，電子計算機を多数の利用者が共同で使用する点では，メッセージを交換するシステムもそうであるし，一般のリアルタイム・システムでも同じである。しかし，これらのシステムでは各利用者はお互いに関連し合っている。たとえばメッセージ交換システムではお互いにメッセージを交換し合う目的があり，3.1.4 の例では，列車の指定券の販売という1つの目的のために多くの端末装置が設置されているのである。

しかし，ここでいう共同利用の場合には，各利用者は必ずしも同じ目的を持つとは限らない。大形電子計算機を共同で使用することのために使用目的において制限が加えられるのであれば，むしろ自由に使える小形の電子計算機を別個に持った方がよいかもしれないのである。小形の電子計算機を使用する代わりとして大形の電子計算機を使用するのであるから，各利用者から見てあたかも自分の電子計算機を使用するように自由に利用できるのでなければ意味がない。このようなことの可能なシステム，すなわち，多くの利用者が共同でかつ独立して電子計算機を利用できるようなシステムを**タイムシェアリング・システム**という。

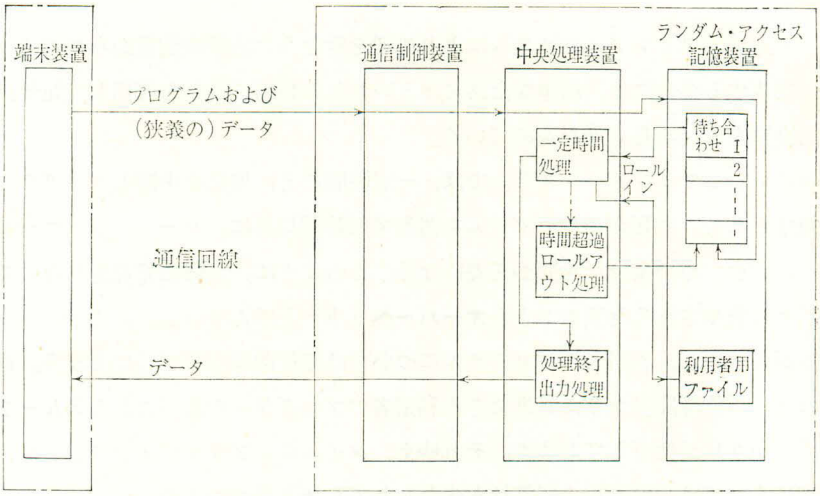
（1） cost performance　（2） Grosch's low



第3.9図　タイムシェアリング・システムにおけるプログラムの処理順序

タイムシェアリング・システムでは，各利用者は自分のプログラムも送ることができる。利用者A，B，Cから送られてきたプログラムをプログラムA，B，Cとし，これ以外に処理すべきプログラムがなかったとする。**第3.9図**に示すように，到着順にしたがってまずプログラムAが処理される。タイムシェアリング・システムでは，1つのプログラムをつづけて処理する時間を，たとえば0.5秒というように制限している。もし，プログラムAが0.5秒以内に終わらなければ，そこで一度プログラムAの処理を中止し，待ち合わせの一番最後にならばせる。つぎにプログラムBが割り当てられた0.5秒だけ処理され，そのあと待ち合わせの最後にまわされる。プログラムCは0.5秒以内に処理を終わってしまうので，その結果はすぐに利用者Cに送り返えされる。

もし各プログラムを処理する時間に制限がなければ，プログラムCはプログラム



第3.10図　タイムシェアリング・システムの例

A，Bが終わるまで待たなければならない。そして，プログラムA，Bが長時間の処理を要すると，処理の短くてよいはずのプログラムCはたまたま到着順が少し遅かったために長い間待たなければならないが，タイムシェアリング・システムでは処理される機会が対等に回ってくる点が改良されている。なお，第3.9図の（5）で，プログラムBの割り当て時間が終了したとき，つぎに待ち合わせているプログラムがなければ，引き続づいてプログラムBの処理が行なわれる。

第3.10図はタイムシェアリング・システムの動作を示したものである。端末装置から送られてきたプログラムは，まずランダム・アクセス記憶装置に記憶され順番を待ち，順番がくると主記憶装置に読み出されて（これを**ロールイン**(1) という），そのプログラムについての処理が行なわれるが，処理に当たってはランダム・アクセス記憶装置のその利用者に割り当てられた範囲を自由に使用することができる。

一定時間過ぎると，このプログラムは再びランダム・アクセス記憶装置に移し換えられ（これを**ロールアウト**(2) という），代わりにつぎの順番のプログラムが主記憶装置に読み出されてくる。処理が終了した場合には，その結果が端末装置へ送り

（1） role in　（2） role out

返される。

タイムシェアリング・システムにより処理を行なうため端末装置からプログラム等を投入する場合には，簡単な会話言語といわれる特別の言語を使用して電子計算機と会話をしながら処理を進めていく。

タイムシェアリング・システムでは，一定時間ごとに処理を中断しプログラムを交換するので，利用者のプログラムを処理する時間以外に，ロールイン－ロールアウトなどのための余分な時間が必要になる。このように，実際に処理を行なっている以外に費やされる時間のことを**オーバーヘッド**(1)と呼んでいる。このオーバーヘッドが大きいと，利用者のプログラムについては処理効率が下がってしまう。換言すれば，単位時間当たりに処理できる利用者のプログラムの数（これを**スループット**(2)という）が低下してしまう。それゆえ，タイムシェアリング・システムは，使い方によっては共同利用の経済性が失なわれてしまう場合がある。

たとえば結果をすぐに手に入れる必要がない場合は，3.1.2で述べたリモートバッチ処理による方が電子計算機の使い方としてははるかに効率的であるし，また入出力するデータは多いが処理が比較的簡単な場合には，このような多量のデータを遠隔地から伝送することを含めて考えると，小形の電子計算機を個別に持っていた方がかえって有利な場合が出てくることも想定される。

しかしタイムシェアリング・システムでは，遠隔地にいる個々の利用者が自分でプログラムを書き，そのプログラムを電子計算機にかけ，誤りがあればそれを修正してすぐその場でためしてみるという操作を行なうことができるので，正しいプログラムを作り上げるに要する時間を格段に縮めることができる。このようにタイムシェアリング・システムは，リアルタイム処理を必要としかつ入出力データの少ない業務に適しているといえる。

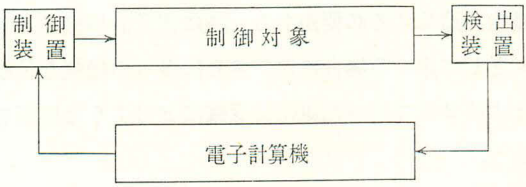
3.1.6　その他のシステム

第3.11図（a）に示す例は，遠隔地にある情報発生地点からのデータを集め，センタの電子計算機により処理を行なうが，処理結果を返送する必要のないシステムで，**データ収集システム**といわれる。

（1）　over head　　（2）　throughput

(a) 情報収集システム

(b) 情報分配システム



(c) 自動制御システム

第3.11図 データ収集分配システムと自動制御システム

図(b) の例は，逆にセンタで処理した結果を，それを必要とする遠隔地に送り出すシステムで，**データ分配システム**と呼ばれる。

いずれもセンタでの処理は必ずしもリアルタイムではない。また，データの収集分配も時々刻々行なわれるとは限らず，電子計算機内の**時計**を利用して周期的にデータの収集，分配を行なったり，あるいはデータが一定量まとまるのを待って一括処理的に行なったりしている。

図(c) の自動制御システムはプロセス制御システムとも呼ばれ，電子計算機を利用して自動車の組み立て工程の制御，交通制御，電力の供給制御等の自動制御を行なうものであり，個々の生産工程を制御する小規模なシステムから，大規模なものでは全工場の生産計画を立てるといった，どちらかといえば 3.1.4 で述べたシステ

ムに近いものもある。このシステムには高い信頼性と，対象によって即時性が要求される。

自動制御システムで，工程の進行状況や制御に必要なデータを各種の計器から人間が読み取ってそのデータを投入し，電子計算機が適当な処理を行なって必要な指示を人間に与え，人間がその工程や所要の装置の操作を行なう場合を**オープンループ**(1)といい，人間が介在せず機械が直接状態を検出し，処理を行なった後必要な操作を自動的に行なう場合を**クローズドループ**(2)という。

3.2　高性能の電子計算機の機能

第2章で述べた電子計算機でももちろん一通りの処理をこなすことができるが，能率の点ではまだまだ不十分な点があり，より効率的なデータの処理を行なうためにはいくつかの機能を追加する必要がある。特に電子計算機をオンライン・リアルタイム・システムとして用いる場合には，さらに多くの機能を付与しなければならない。そこで，まず第2章で述べた電子計算機にどのような機能を追加すればよいかを考察してみよう。

3.2.1　高性能の電子計算機に対する要求

データ処理の効率を上げるためにまず問題となるのは，入出力装置の速度のおそさをいかに補って処理を進めるかという点である。中央処理装置がどんなに高速の論理回路を持っていても，低速の入出力装置をじようずに駆使できなくては，その機能を十分生かすことができない。この問題を解決する方法として，第2章で述べたようにチャネルの概念が生まれた。すなわち，個々の入出力装置の制御をチャネルに行なわせることにより，複数の入出力装置を同時に動作させることが可能になった。しかし，実際にはこれでも不十分なのである。

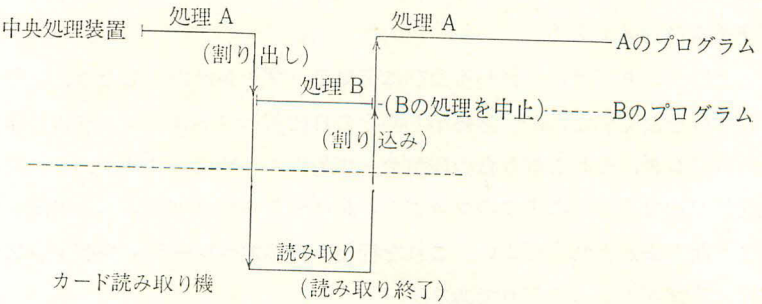
第2章ではチャネルの説明として，行印字機に出力している間に，つぎのデータの計算を行なう例を示した。この場合，たまたま計算すべき仕事があればよいが，もし何もする仕事がなければ，中央処理装置はやはり遊んでしまうことになる。そこ

(1)　open loop　　(2)　closed loop

で，この間を利用してまったく別のプログラムの処理を行なおうという考えが生れた。この思想をさらに押し進めていくと，1台の計算機が多数のプログラムをそれぞれのプログラムが持つ間隙を適当に利用しながら処理を進め，見かけ上これらのプログラムが同時に処理されているようにすることができる。このように複数のプログラムからプログラムを適当に選び出しながら処理を進めていく方法を，**多重プログラミング**(1)と呼んでいる。

第3.12図を用いてさらに具体的に説明すると，電子計算機がプログラムAの処理を進めていくうち，「カード読み取り機からデータを読み込め」という命令にぶつかったとする。チャネルにカード読み取り機の制御を行なわせることで，中央処理装置はつぎの処理を進める状態になっているのであるが，ここでプログラムAにはデータの読み取りが終了するまで処理すべきものがない場合（たとえば，カード読み取り機から読み込んだデータによってつぎの計算をするようになっている場合）には，データの読み取りが完了するまでの待ちの時間を利用して，別のプログラムBを実行しようというのである。このように，待ちの状態にはいったプログラムの処理から別のプログラムの処理へ移ることを**割り出し**(2)という

さてデータの読み取りが終了すると，終了したという信号が中央処理装置に送られてくる。これによってプログラムBの処理は強制的に中止され，再びプログラムAの処理にもどされる。これを**割り込み**(3)という（この割り込みと先の割り出しを



第3.12図　割　り　込　み

（1）　multiprogramming　（2）　supervisor call　（3）　inturruption trap

合わせて，割り込みという場合がある)。

多重プログラミングを行なうためには，上述の割り込み機能が第一に必要なものであるが，そのほか主記憶装置のアドレスを各プログラムに割り当てるための機能や，あるプログラムを実行中に他のプログラムが使用している記憶内容が破壊されないように保護する機能などが必要になってくる。これらについては後でさらに詳細に説明する。

処理速度を上げる上でつぎに問題になるのは，主記憶装置からのデータの読み出し書き込み時間である。これを向上するのには主記憶装置の高速化をはかればよいわけであるが，もっと簡単な方法としては，同時に多数のデータを読み書きしたり，主記憶装置を幾つかのコア・スタックに分け，1つのコア・スタックからの読み取りを行なった後，つぎの読み取りが可能となるまでの間，別のコア・スタックからの読み取りを行なう，いわゆる**メモリー・インターリーブ**[1]などの方法がある。

制御装置そのものについても，効率化をはかる手法が考えられている。たとえば**先回り制御**もその1つである。この方法は1つの命令を読み取り，その命令を実行し終わるまでの間につぎの命令を先回りして読み取っておくものである。さらに，中央処理装置を複数台組み合わせて並列に処理を行なわせ効率化をはかることも考えられている。

もう1つ電子計算機の効率を上げる上で忘れてならないことは，磁気テープをかけ換えたり，装置のボタンを押したりする操作のための時間，すなわち，**ハンドリングタイム**の削減である。

たとえば，プログラムが終わるたびに磁気テープをかけなおしたり，装置の操作をする手間をなくすために，あらかじめその日に実行すべきプログラムを磁気テープに入れておき，それを複数台の磁気テープ装置にかけておけば，あとは最初のボタン操作だけで自動的に1つのプログラムからつぎのプログラムへと処理が移っていくような方法がとれればよい。これを行なうのが**オペレーティング・システム**[2]と呼ばれるプログラムの集りである。

電子計算機がオンライン・リアルタイム処理に使用される場合には，いつはいっ

(1)　memory interleave　　(2)　operating system

てくるかわからないデータを待ち受けて即時にかつ効率的に処理を行なうために，上述の割り込み，主記憶装置の割り当て，記憶保護などの機能は欠くことのできないものであり，またオペレーティング・システムにも，人間の介在を必要とせずに処理を行なえるような機能が要求される。オンラインであるからには，通信回線を通して端末装置を制御するための通信制御の機能も必要である。さらにリアルタイム・システムに対しては，一般により低い故障率，すなわち高い信頼性が要求される。信頼性に対する要求は個々のシステムによって異なるが，処理能率を低下させることなくいかなる中断も許さないような方式（これを**フェイル・セーフ**という）や，たとえ故障になってもある程度能率が落ちるだけで中断されることのないようにする方式（これを**フェイル・ソフト**[2]という）などがあり，これを実現する手段として複数の電子計算機を組み合わせて使うなどの方法が取られている。

3.2.2 割り込み

広い意味での割り込みとは，あるプログラムの実行中に割り込んで別のプログラムを実行することをいい，高性能の電子計算機ではきわめて重要な機能である。このような動作は，必ずしも前述の多重プログラムの場合に限らず，一般のプログラムの処理を実行中にも現われる。

たとえば，機械の誤動作が発生した場合，もう一度同じ処理を試みるか，あるいは必要な指示を操作員にあおぐため制御卓のタイプライタに障害状況を印字するなどの動作が必要になる。このような場合，いかなる処理を行なわせるかは電子計算機の利用の仕方によって変わるので，ハードウェアで回路に作ってしまうわけにはいかず，プログラムで規定しておくのが普通である。こうして，あるプログラムの実行中に機械の誤動作が発生したとすると，実行中のプログラムを強制的に中断し，機械誤動作時の処理を行なうプログラムに移ることになる。このような動作を，実行中のプログラムが障害処理プログラムにより割り込まれたというようないい方をする。

割り込みはこのほかにも，プログラムの誤りによって使ってはいけない命令が現われたり，主記憶装置の書き込みを禁止されているアドレスに対して書き込む命令

---

(1) fail safe (2) fail soft

が現われたりしたとき，あるいはまた操作員が必要に応じて制御卓から割り込み指示を行なった場合にも生ずる。以上まとめて割り込みの原因を列記すると，

（1）　入出力装置の動作終了に基づく割り込み

（2）　そのプログラムが終了または待ちの状態になったときの割り出し

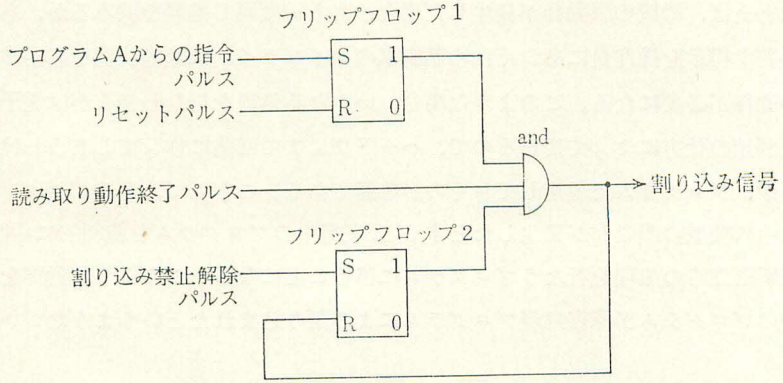
（3）　機械の誤動作や電源異常に基づく割り込み

（4）　プログラムの誤りによる割り込み

（5）　操作員からの要求に基づく割り込み

それでは割り込み動作をもう少し具体的に説明しよう。再び第3.12図を例として取り上げてみる。

割り込み信号(1)は，カード読み取り機が読み取り動作を終了するつど送られるのではなく，あらかじめその旨，指令を受けているときにのみ送られてくる。たとえば第3.12図でプログラムAは割り出し後は読み取り動作の終了を待っている状態にあり，読み取り動作の終了後はただちに割り込み信号を送ってもらう必要がある。そこでプログラムAは，カード読み取り機に対しチャネルを通して動作指令を与える際に，同時に読み取り動作終了後に割り込み信号を送れという命令を与えておかなければならない。



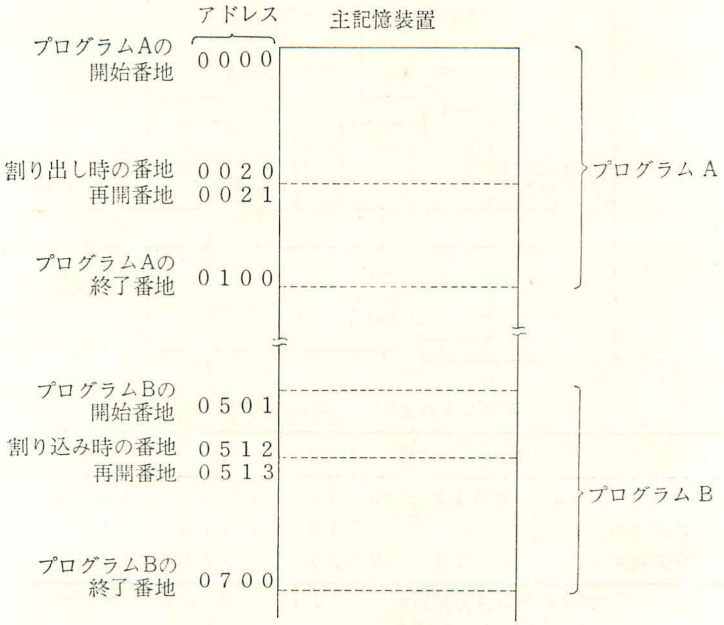
第3.13図　割り込み信号送出回路の原理図

（1）　interrupt signal

第3.13図に割り込み信号送出回路の原理図を示す。上記の命令は，具体的には同図に示すプログラムAからの指令パルスの形で与えられ，フリップフロップ1をセットする。フリップフロップ2は通常セットであるので，カード読み取り機から読み取り動作終了パルスがくると，and ゲートが開いて割り込み信号用のパルスが送り出されることになる。

割り込み処理中につぎの割り込み信号がくると混乱を生ずるため，これを防ぐ意味でフリップフロップ2があり，割り込み処理が終了してつぎの割り込みが受けつけられるようになると，割り込み禁止解除パルスが送られてフリップフロップ2はセットされる。なお，フリップフロップ1は第3.13図には1個のみ示してあるが，実際には各種の割り込み原因について，また，入出力関係の割り込みについては各チャルごとに個々に設けられている。ただし，フリップフロップ2の方は1個あればよい。

さて，割り込み信号が送られてくると，実行中のプログラム（第3.12図ではプロ

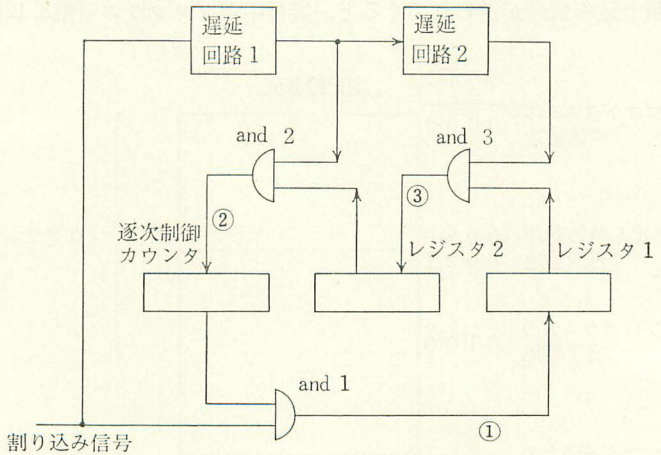


第3.14図　割り込み時の主記憶装置の状況

グラムB）を中断しなければならないが，それに先だって各種のレジスタやフリップフロップに一時的に記憶されている中間結果等が，プログラムBの中断中に別のプログラムによって破壊されないように，レジスタやフリップフロップの内容を主記憶装置や別のレジスタに待避させなければならない。

一例として，プログラムBの再開後の開始番地のはいっている逐次制御カウンタの内容を待避する場合を考える。**第3.14図**で，プログラムBが0512番地まで実行されたところでカード読み取り動作が終了し，割り込み信号が送られてきたとする。プログラムBをここで中断し，プログラムAを 0021 番地から実行しなければならない。それには逐次制御カウンタに現在はいっている0513番地を別の レジスタに待避させ，代わりに0021番地を逐次制御カウンタに入れてやればよい。

第 3.15 図では逐次制御カウンタの内容を待避させるレジスタがレジスタ1，レ



カウンタおよびレジスタの内容

	割り込み寸前	①	②	③
逐次制御カウンタ	0513	0513	0021	0021
レジスタ 1	—	0513	0513	0513
レジスタ 2	0021	0021	0021	0513

プログラムA再開番地　0021
プログラムB再開番地　0513

第3.15図　逐次制御カウンタの内容の待避

ジスタ2になっている。同図で and 1，and 2，and 3 の各 and ゲートは簡単のため1個しか示していないが，実際にはカウンタやレジスタの各ビットごとに設けられており，割り込み信号がくると and 1 ゲートがいっせいに開き，逐次制御カウンタの内容がそっくりレジスタ1に転送される。レジスタ2には，割り込みによって実行さるべきプログラムここではプログラムAのアドレス 0021 がはいっており，割り込み信号が遅延回路1により遅らされて逐次制御カウンタの内容がレジスタ1に転送されたあとで and 2 ゲートに加えられると，逐次制御カウンタにそっくり転送される。つぎに割り込み信号は遅延回路2によりさらに遅らされて，レジスタ2の内容が逐次制御カウンタに転送されたあとで and 3 ゲートに加えられ，レジスタ1の内容がレジスタ2に転送されて，プログラムBが再開されるのを待つ。

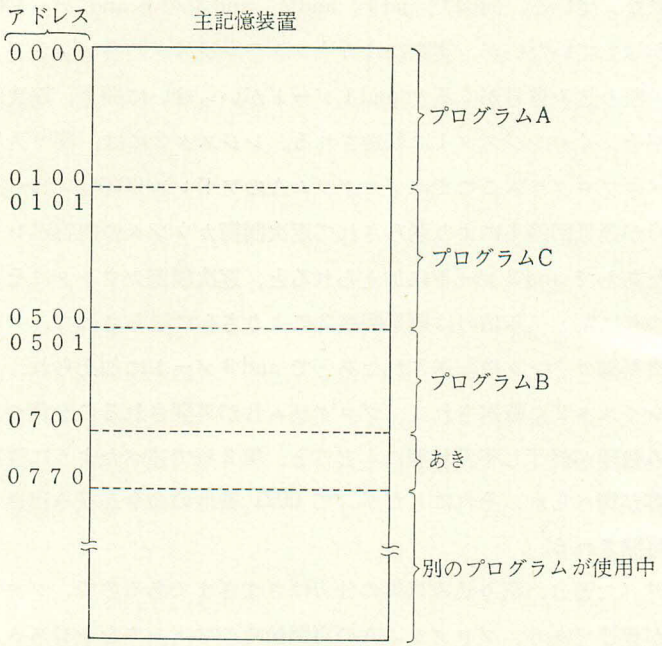
割り込み処理が終了し平常状態にもどると，第2章で述べたように逐次制御カウンタの内容が調べられ，それにしたがって 0021 番地の命令が読み出されてプログラムAが再開される。

前にも述べた通り，割り込み処理の仕方はさまざまであるので，プログラムで規定するのが普通であり，プログラムAの再開始時のアドレスをレジスタ2に入れておく代わりに，割り込み処理用のプログラムのアドレスを入れておく。割り込みが発生すると，割り込み処理用のプログラムが実行され，割り込み原因が分析されて，各々の原因に応じた処理が行なわれる。割り込み処理が終了すると，第3.13図のフリップフロップ1をリセットし，さらに割り込み禁止解除パルスを送ってフリップフロップ2をセットする。

割り出しの場合には，そのプログラムでつぎに実行すべきプログラムのアドレスを直接指定することも可能であるが，割り込み回路を利用して行なうのが普通である。

3.2.3 主記憶装置の割り当て

電子計算機がただ1つのプログラムを処理する場合には，主記憶装置のいずれのアドレスでも自由に利用してプログラムの処理を行なうことができる。しかし，多重プログラミングの場合のように，複数のプログラムが主記憶装置を使用する場合には，**主記憶装置の割り当て**ということが必要になってくる。

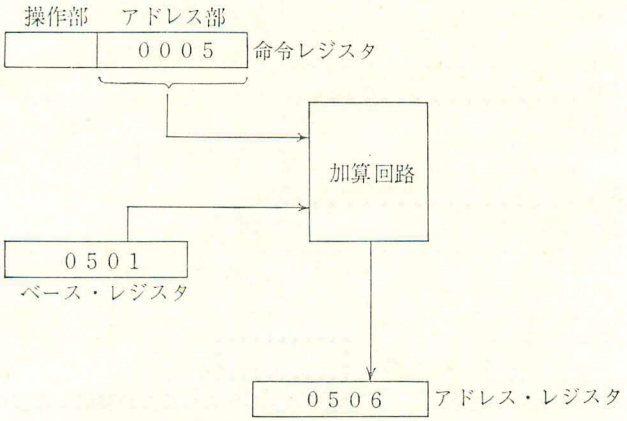


第3.16図　主記憶装置の割り当て

たとえば，**第3.16図**に示すように，0001番地～0100番地がプログラムAの処理のためにすでに使用されている場合には，この範囲のアドレスはもはやプログラムBの処理に利用することはできない。しかし，0001番地～0100番地を他のプログラムが使用しているかどうかはそのときになって始めてわかるので，プログラムを作成している段階では予知できない。そこで複数のプログラムの処理を行なう場合には，それぞれのプログラムが使用できる主記憶装置のアドレスの割り当てをそのつど決める必要がある。

いま，プログラムA，B，Cを主記憶装置に記憶させる必要があったとし，各々第3.16図に示すアドレスを割り当てたとする。各プログラムは，主記憶装置が他のプログラムによりいかに使用されているかには無関係に，それぞれ主記憶装置の0番地から使用できるものとして書かれている。たとえばプログラムBの命令の中に，アドレス部が0005になっているものがあったとする。これをそのまま実行する

とプログラムAの中の命令（またはデータ）を読み出してしまう。これを防ぐためには，プログラムBを実行中はアドレスを 0501 番地だけずらしてやればよい。すなわち**第 3.17 図**に示すように，そのプログラムが使用しうる最初のアドレス（ここでは 0501）をレジスタに入れておき，これとプログラムで指定したアドレス（ここでは 0005）との加算を行なって，その結果（ここでは 0506）を主記憶装置のアドレスとする。

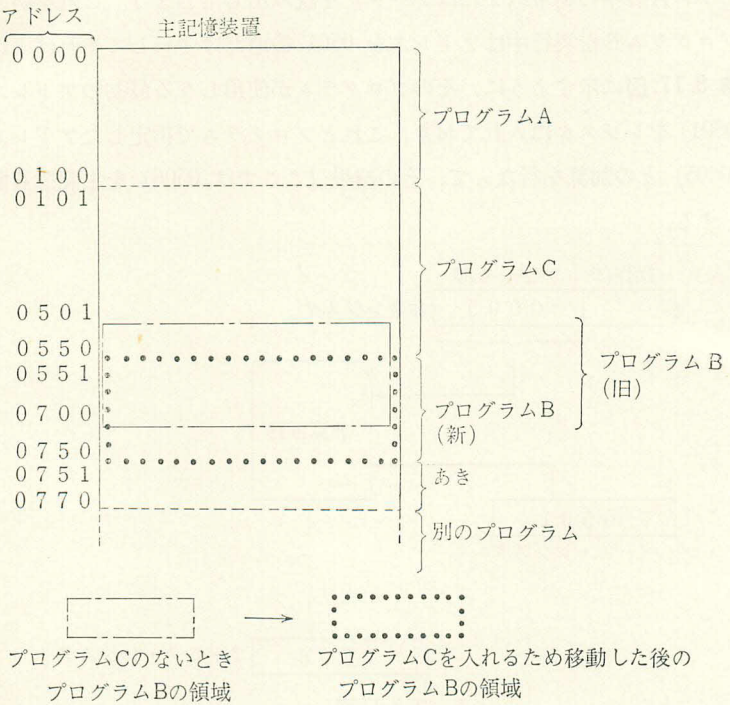


第 3.17 図

この場合，0506 のように主記憶装置に固有のアドレスを **絶対アドレス**(1) といい，0005 のようにプログラミングに際して相対的に付されたアドレスを**相対アドレス**(2)という。また，相対アドレスをづらして絶対アドレスに変換するために基準となるアドレス（ここでは 0501）を記憶させておくレジスタを **ベース・レジスタ**(3) という。

つぎに，第 3.16 図においてプログラムCが大きく，0101 番地から割り当てたとすると 0550 番地まで必要になるとする。この場合，0501 番地から 0550 番地の範囲でプログラムBとプログラムCが重複するのを避けるためには，**第 3.18 図**に示すように，プログラムCにアドレスを割り当てるに先立って，0501 番地から 0700 番地に記憶されているプログラムBを 0551～0750 番地に移し変えるとともに，プログ

（1） absolute address （2） relative address （3） base register



第3.18図　ダイナミック・リロケーション

ラムBを実行する際にベース・レジスタに 0551 を入れるようにする必要がある。このように，プログラムの実行が終了する前にプログラムの収容位置を変えることをプログラムの**動的再配置（ダイナミック・リロケーション）**(1)という。

オンライン・リアルタイム・システムでは，端末装置から無作為に送られてくるデータに応じて処理を行なうプログラムがつぎつぎと切り換えられるので，プログラムの入れ換えを行なうための動的再配置の機能が必要である。

プログラムCがさらに大きく，プログラムBを動かしても主記憶装置に収容できない場合には，プログラムCの一部分だけを主記憶装置に収容し他の部分は補助記憶装置に記憶させておき，主記憶装置に収容されている部分の処理が終了した

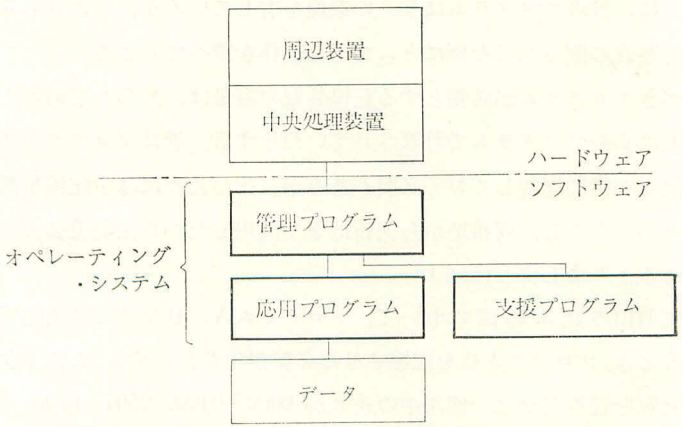
(1) dynamic relocation

ら，その部分と補助記憶装置に記憶されている残りの部分とを移し変えて処理を続行する方法が取られる。このような処理も動的再配置の機能により可能となっている。

3.2.4 オペレーティング・システム

3.2.3 においては，各プログラムをだれが主記憶装置に割り当て，だれがベース・レジスタにアドレスを入れるのか述べなかった。この**だれ**の正体は，**オペレーティング・システム** (OS)[1] といわれるプログラムの集りである。

オペレーティング・システムは，もともと電子計算機の操作員を助けるため，電子計算機ができるだけ自動的に処理を行なえるようにしようという思想から生れたものであった。しかし，高性能の電子計算機では，操作員が介入することにより処理が中断される機会を減らし，電子計算機の能力を最大限に活用しようという目的に変わってきている。特にオンライン・リアルタイム・システムでは，本来，操作員の操作は必要最小限に押さえられており，操作員を助ける目的よりは，多数の端末装置から無差別にはいってくるデータを受け，多数のプログラムをそのときの状況に応じて使いわけながら，効率的な処理を行なうためのプログラムになっている。オペレーティング・システムの構成を第 3.19 図に示す。



第 3.19 図　オペレーティング・システム

(1)　operating system：略して OS

管理プログラムは，オペレーティング・システムの中でももっとも重要なプログラムであり，電子計算機のハードウェアと密着して基本的な処理を行なうものである。せまい意味では，この管理プログラムのことをオペレーティング・システムという場合もある。また，管理プログラムは，制御プログラム，主制御プログラム，モニタなど各種の名称で呼ばれている。

支援プログラムは，ユーティリティ，サービスプログラムなどとも呼ばれ，ソース・プログラムをオブジェクト・プログラムに変換するプログラム（コンパイラ，アセンブラなど）や分類作業を行なわせるためのプログラムなど，電子計算機の利用者にとって有用なプログラムからなっている。

応用プログラムは，作業用プログラム，処理プログラムなどとも呼ばれ，実際にデータ（せまい意味で）を処理するプログラムである。

管理プログラムは周辺装置を制御する機能を有している。すなわち，チェネルを通して各周辺装置に指示を与えるのは，個々の応用プログラムではなく管理プログラムが行なっている。また，各種のプログラムの処理される順序を決めたり，入出力装置の割り当てを行なったり，あるいはまた，3.2.2で述べた割り込み処理を行なったりするのも，すべて管理プログラムである。

このように，管理プログラムは多くの機能を有しているが，そのうち3.2.3で述べた主記憶装置の割り当てを例にとってその動作を調べてみよう。

実行すべきプログラムが必要とする記憶装置の容量は，あらかじめ与えられているか，または別のプログラムで計算されていたとする。管理プログラムは主記憶装置の使用状況を常に理解しておく必要があるが，そのためには主記憶装置をプログラムに割り当てるつど，何番地から何番地まで使用したかを主記憶装置の適当な場所に記憶するようにしておけばよい。

たとえば前出の第3.16図において，プログラムA，Bがすでに主記憶装置を使用しているとき，プログラムCを記憶させる必要が生じたとする。このとき上記の記憶場所の内容を読み出せば，使用中の番地が 0000～0100，0501～0700，0770～××××であることがわかる。0501と0100，0770と0700の差を取ることにより主記憶装置のあいている場所の大きさがわかるので，それらの大きさとプログラムC

の大きさとを比較し，プログラムCが0100と0501番地の間にはいるならばそこに割り当てる。第3.18図のようにプログラムCの大きさがあいている場所よりも大きい場合には，プログラムBの移し換えを行なうことになる。

管理プログラムは上述のようにきわめて重要な機能を有しており，そのため特殊な命令（**特権命令**とか**主制御命令**とかいう）も使用している。これらの命令は，もし一般のプログラムに使用されたりすると，そのプログラムの処理中に他のプログラムの内容を破壊してしまうなどの不測の事態を生ずるおそれがある。そこで管理プログラム処理中と一般のプログラムの処理中とを区別し，一般のプログラム処理中にこれらの命令が出てくると割り込みが発生するようになっている。

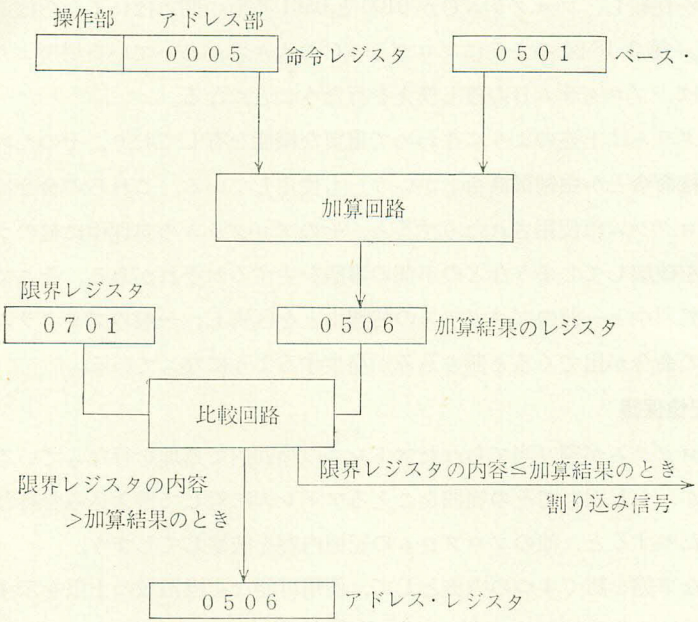
3.2.5 記憶保護

個々のプログラムが割り当てられたアドレスの範囲内で処理を行なっているうちは問題ないが，もし誤ってその範囲をこえるアドレスに対して書き込みを行なう命令が出されたりすると，他のプログラムの記憶内容を破壊してしまう。

このような事態を防ぐ1つの方法として，使用可能な記憶領域の上限を示す限界レジスタを設ける方法がある。すなわち，**第3.20図**において，命令のアドレス部とベース・レジスタの内容を加算した結果を限界レジスタの内容と比較し，それよりも小さければそのままアドレス・レジスタに送り処理を進めるが，限界レジスタの値をこえた場合は割り込みを生じ，管理プログラムによる必要な処理がなされる。

記憶を保護するもう1つの方法では，主記憶装置を幾つかの領域に分け，各々の領域に1ビットを対応させておく。もしその領域を他のプログラムが使用しているときにはそのビットを“1”に，使用していないときには“0”にする。そしてビットが“1”になっている領域に対して書き込みの命令が出された場合には，前と同様割り込みを起こさせればよい。

以上は書き込みに対する記憶の保護の例であるが，多数の利用者が同一の電子計算機を使用する場合には，ある利用者の記憶内容を他の利用者に読み取られることのないようにする機密の保護が必要になる。このように読み取りに対しても制限を行なう場合には，主記憶装置の領域を自由使用領域，書き込みの不能な領域，書き込み読み込みとも不能の領域に分けて制御を行なわなければならない。



第3.20図　記憶保護回路

機密保護を行なうには，しばしば**合鍵方式**(1)が用いられる。これは主記憶装置の各領域に幾つかのビットの組み合わせを与えておき，各プログラム(または利用者)に与えられた**合鍵**のビット構成とその領域に与えられたビット構成が同じなら読み取りも可能にするというものである。

3.2.6　通信制御

電子計算機をオンライン・システムに使用する場合には，通信回線を通して端末装置を制御する機能を特別に付加しなければならない。これを通信制御という。通信制御を中央処理装置で行なわせることも不可能ではないが，そのために中央処理装置の能力の一部がさかれるのは得策ではなく，通信制御に専任する通信制御装置を別に設けているのが普通である。

電子計算機と端末装置との間でデータを送受信する場合には，つぎのような通信

(1)　pass word 方式

制御が必要である。

（1） 端末装置からのデータを受信する場合には，データの送出を希望する端末装置の存在を判定し，相手を確認して，必要あらば受信準備完了信号をその端末装置に送る。また，端末装置へデータを送信する場合には，相手を呼び出し，受信可能かどうかを確認する。

（2） 端末装置から送られてきたデータを受信し，電子計算機の中で処理しやすい形，すなわち，内部符号に変換して記憶する。また，端末装置へデータを送信する場合には，内部符号を伝送用の符号に変換して通信回線に送出する。

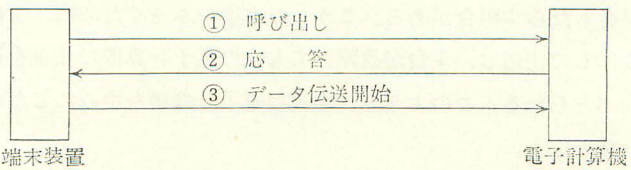
（3） データが正しく受信されたかどうかを調べ，誤りが存在するときは必要な処理を行なう。たとえば相手の端末装置に対して「もう一度同じデータを送れ」という信号を送る。

（4） データの送受信の終了にともない，その端末装置を切り放し，他の端末装置との間でデータの送受信をする準備を行なう。

ここで，(1)の相手の状態を確認する手順について説明する。

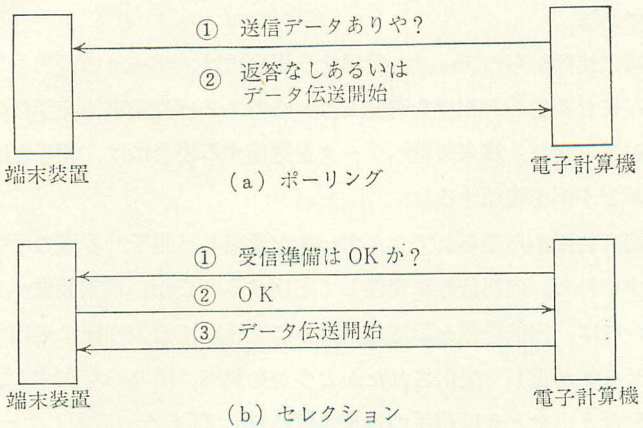
第3.21図は，**呼び出し応答方式**(1) といわれる方法で，まず送信しようとする側から呼び出し信号を送り，それに対して受信側では準備ができていれば応答信号を送り返す。送信側は応答信号を確認してからデータの伝送を開始する。呼び出し応答を行なう際に，併わせて相手先と自分の識別コードを送ることにより，正しい相手に接続されたかまたはどの相手から接続されたかを判定できる。

第3.22図は**ポーリング・セレクション**(2) といわれる方法である。この場合には，電子計算機がすべての制御を行なう。電子計算機から端末装置に対して送信データ



第3.21図 呼び出し応答方式

（1） contention （2） polling selection



第3.22図　ポーリング・セレクション方式

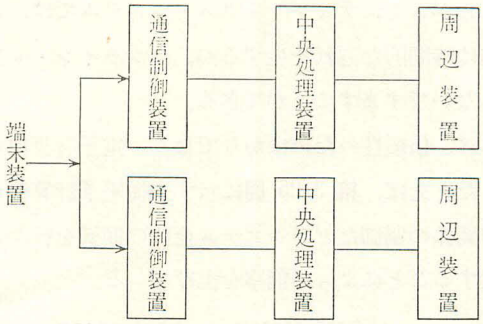
の有無の問い合わせを行ない(これをポーリングという)，もし該当装置が送信すべきデータを持たないときは応答を行なわないので電子計算機はつぎの端末装置に対して送信の問い合わせをする。電子計算機よりデータを送信する場合は，受信準備が完了しているか否かの問い合わせ（これをアドレッシングまたはセレクションという）を行ない，準備完了の返事が得られるとデータの送信を開始する。

3.2.7　電子計算機複合体

いままで述べてきた電子計算機では，中央処理装置1台とそれを中心にした周辺装置により構成されていた。この場合には，中央処理装置が故障になると電子計算機はその機能をまったく停止してしまう。一括処理のシステムでは，電子計算機が故障している時間だけ処理が遅れるだけですむが，オンライン・リアルタイム・システムでは電子計算機の故障により工場全体がストップしてしまうといったような重大な結果をもたらす場合がある。こうした事態をふせぐために，複数台の電子計算機を組みにして使用し，1台が故障しても他の電子計算機で仕事を進めていくといった方法がとられる。このように，複数の電子計算機を組みにしたものを電子計算機の複合体という。

第3.23図に示す例は**デュアル・システム**(1)と呼ばれ，2台の電子計算機で同一

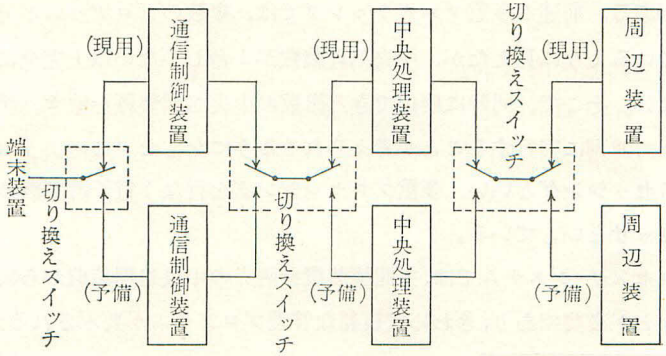
(1)　dual system



第3.23図 デュアル・システム

の処理を実行し，互いに相手と照合をとりながら処理を進めていくものである。両方の電子計算機が同時に故障した場合にはシステムとしての機能を失うが，このようなことは確率的にきわめてまれにしか起きないので，高い信頼性を得ることができる。絶対に故障を起こしてはならないシステム（フェイルセーフ・システム）にはデュアル・システムが使用される。

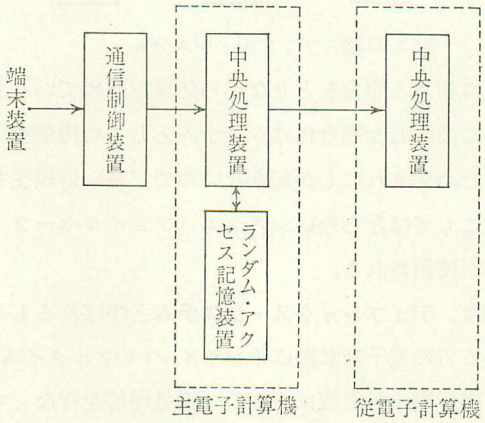
第 3.24 図に示す例は，**デュプレックス・システム**と呼ばれるもので，正常の状態では**現用**と書いてある方の電子計算機はオンライン・リアルタイム処理を行ない，一方**予備**と書いてある方の電子計算機は，別の一括処理等を行なっている。現用側の電子計算機が故障した場合には，予備側の電子計算機はそれまで行なっていた一括処理をただちに中断し，現用機が行なっていたオンライン・リアルタイム処理を



第3.24図 デュプレックス・システム

継続して行なう。したがって，デュプレックス・システムでは，電子計算機の故障により一括処理業務は時間的な遅れを生ずるが，オンライン・リアルタイム処理には大きな影響を与えないですますことができる。

電子計算機複合体は，信頼性のためばかりでなく，電子計算機の効率を高める目的でも使用される。たとえば，**第 3.25 図**に示す主従電子計算機システムでは，主電子計算機に入出力関係の制御などのシステム全体の制御を行なわせ，従電子計算機を計算専門に使用することによって能率を上げている。

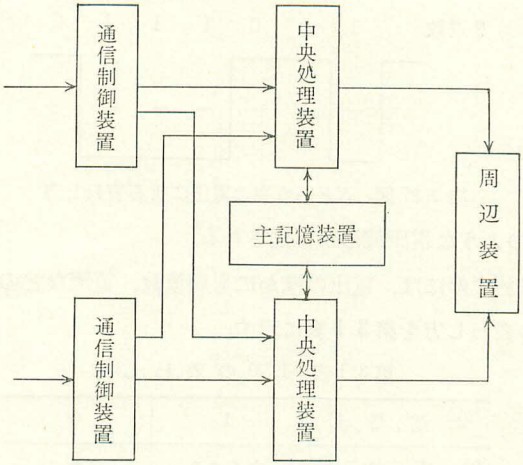


第 3.25 図　主従電子計算機システム

処理の効率をさらに上げる方法として，**第 3.26 図** に示す**多重プロセッサ・システム**(1)がある。前述の多重プログラミングでは，複数のプログラムが見かけ上並列に走っているように見えたが，中央処理装置が1台しかない以上完全に並列には処理できない。そこで，同時に使用できる複数の中央処理装置を置き，幾つかの処理を分担して並列に実行することが考えられるようになった。このような処理方法を**多重プロセッシング**といい，多重プロセッシングを行なう電子計算機の複合体を**多重プロセッサ**といっている。

多重プロセッサ・システムでは，主記憶装置を両方の中央処理装置から読み書きできる機能などが必要であり，きわめて複雑な管理プログラムが要求されるが，処理の

(1)　multiprocessor system



第3.26図　主多重プロセッサ・システム

高速化のためにはきわめて有力な方式である。またオンライン・システムのように多数の端末装置が接続されている場合にも適しており，今後の発展が期待されている。

3.3 データ伝送[1]

3.3.1 データ伝送の方式

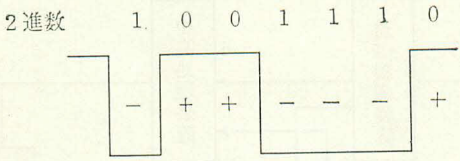
1. データ伝送用の符号　第1章において，電子計算機の内部では数値が2進法で表わされていることを説明した。また第2章では，数値にかぎらずカナやアルファベットなどの文字もすべて2進符号で表わされていることを述べた。データ伝送においても文字や数字を伝送するのに通常2進符号を用いて伝送している。いま，アルファベットの N が7けたの2進符号により，

$$N=1001110 \qquad (3.1)$$

と表わされているものとする。データ伝送では，この2進符号を，電圧の極性，周波数の高低，電流の有無などに置き換えて送る。たとえば“1”を表わすのにマイナスの電圧[2]を，“0”を表わすのにプラスの電圧を送ることに約束すれば，たと

(1) data transmission

(2) 第1章では“1”をプラスで，“0”をマイナスで表わしたが，データ伝送では逆に“1”をマイナスで，“0”をプラスで表わすのが普通なので，この章ではそれに従った



第3.27図　Nという字の電圧による表わし方

えば**第3.27図**のような電圧波形が伝送される。

2進符号を表わすのには，電圧のほかにも周波数，電流などのいろいろな方法がある。これらの表わし方を**第3.1表**に示す。

第3.1表　1，0の表わし方

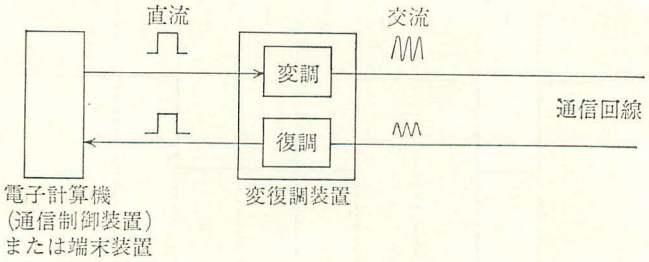
2進符号	1	0
電圧極性	マイナス	プラス
周波数	低い周波数	高い周波数
紙テープ，カード	孔あり	孔なし
トランジスタ	順	逆
ランプ	点火	滅火
スイッチ	オン	オフ

〔注〕 電圧極性は1を正とする場合もある

2. 直流伝送と交流伝送　電子計算機や端末装置の中では，2進符号は直流電圧の形で表わされている。そこで，これらの装置からデータを送り出す場合に，そのまま直流の形で送り出すのがもっとも簡単である。このように，伝送すべき2進符号を直流波形のまま伝送する方式を**直流伝送**，あるいは**ベースバンド伝送**という。直流伝送は一般に雑音に弱く，また，直流信号の増幅が難かしいため，通常距離が短い場合にのみ適用される。

距離の長い場合には，増幅が容易で比較的雑音に強い交流の形でデータ伝送が行なわれる。このためには電子計算機や端末装置の出力端において直流を交流に変換し，また，逆に伝送されてきた交流波形を直流に変換する必要がある。このように，もとの波形を伝送しやすい波形に変えることを**変調**(1)といい，変調された波形をもとの波形にもどすことを**復調**(2)という。また，変調と復調を行なう装置を**変復**

(1)　modulation　(2)　demodulation

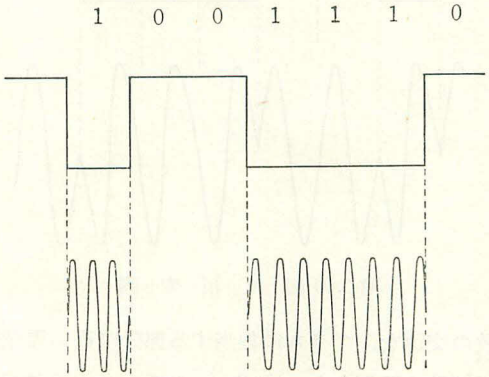


第3.28図 変復調装置

調装置という。変復調装置は第3.28図に示すように，電子計算機（通信制御装置）または端末装置と通信回線の間に置かれる。

3. 変調方式 データ伝送において使用される変調方式のうち，代表的なものについて説明する。

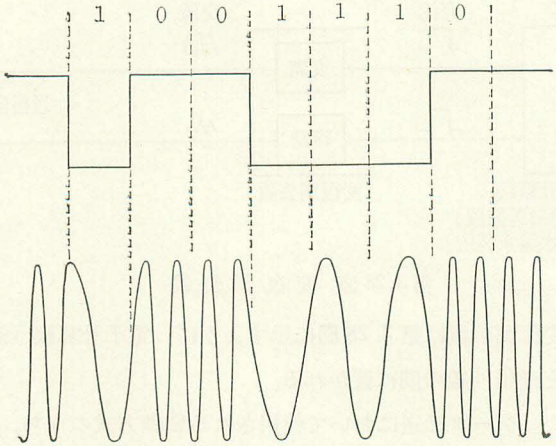
a. 振幅変調(1) 第3.29図に示すように直流波形のマイナスに相当する部分，すなわち“1”のときは交流電流が流れ，プラスに相当する部分，すなわち，“0”のときは電流が流れない方式であり，変復調装置は簡単であるが受信した交流信号が小さなとき“1”か“0”かを判定するのがむずかしい難点がある。



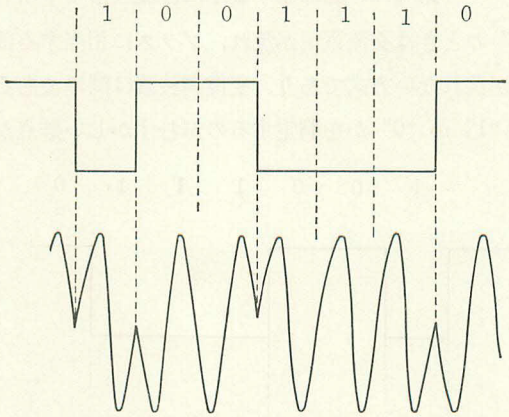
第3.29図 振幅変調

b. 周波数変調(2) 第3.30図に示すように，直流波形のマイナスに相当す

(1) amplitude modulation (2) frequency modulation



第3.30図　周波数変調



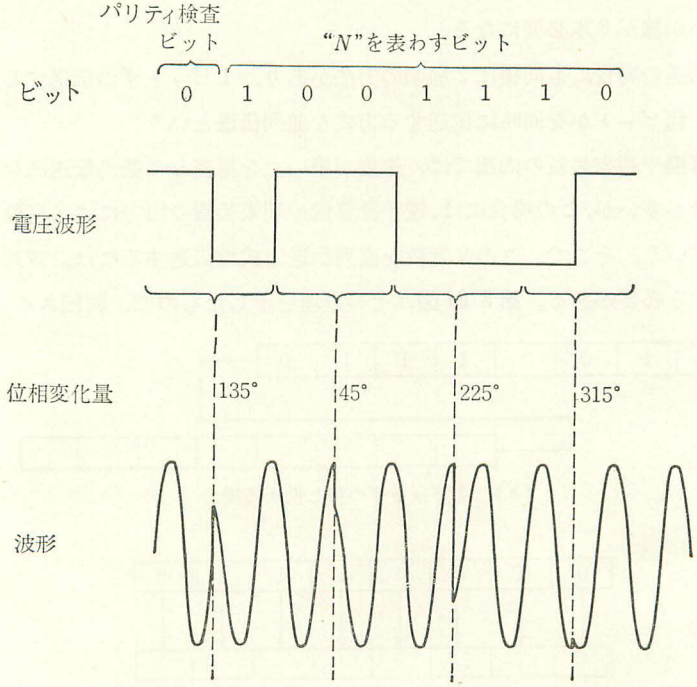
第3.31図　位相変調

る部分は低い周波数の交流を，プラスに相当する部分は高い周波数の交流を送る方式である。2つの周波数の区別さえつけばよいので，多少の雑音があったりまた受信した交流信号が小さい場合であっても“1”か“0”かの判定は振幅変調より容易であり，データ伝送においてもっとも一般的に使用される方式である。

c.　**位相変調**　　直流波形のマイナス，プラスに応じて交流波形の位相を変化

させる方式で，第 3.31 図の例では 180 度位相を変化させている。位相変調では第 3.32 図に示すように，位相を 4 種類に変化させる四相位相変調方式もよく用いられる。

四相位相変調により 2 進符号を伝送するには，2 進数を 4 進数に変換してやる必要がある。すなわち，4 進数の 1 けたが表わしうる 4 種類の数 (0, 1, 2, 3) を 4 つの位相に対応させるのであるが，対応のさせ方にもいろいろな方法がある。たとえば第 3.32 図の例では 01001110 という 2 進数を 01, 00, 11, 10, に区切り，そして各々すぐ前に送った交流信号の位相を基準にして，つぎのように位相をずらした交流信号を送る。



第 3.32 図　四位相変調方式の例

00…… 45° ずらす　　01……135° ずらす

11……225° ずらす　　10……315° ずらす

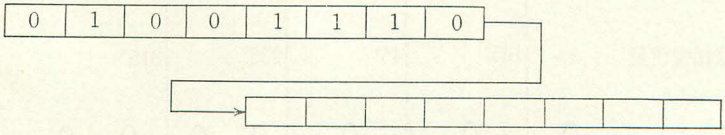
受信側では送られてきた交流の位相を調べ位相の変化量を検出して 00, 01, 11, 10 のどれに当たるかを判定する。

この方法では，1回の位相変化で2進数2けた，すなわち，2ビットを送ることができる。このように四相位相変調方式では，1本の通信回線を用いて単位時間により多くのデータを送ることができるが，装置は複雑となり，また，通信回線にも高い品質が要求される。

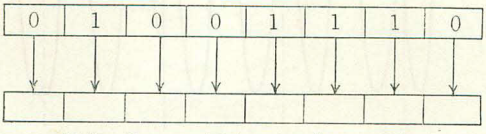
4. **直列伝送と並列伝送**　第 3.33 図に示すように8けたの2進符号，すなわち，8ビットのデータを移し換える場合に，1ビットずつ移し換える**直列転送法**と8ビット同時に移し換える**並列転送法**と2種類あり，後者の場合の方が移し換える時間は短いが線が8本必要になる。

データ伝送の場合にも同様に2種類の方法があり，1ビットずつ伝送する方式を**直列伝送**，何ビットかを同時に伝送する方式を**並列伝送**という。

電子計算機や端末装置の内部では，速度が早い点を重視して並列転送法を採用している場合が多いが，この場合には，電子計算機や端末装置の出力には2進数が並列に現われている。そこで，この2進数を直列伝送方式で伝送するには，並列→直列の変換をする必要がある。**第 3.34 図**はその原理を示したもので，回転スイッチ(実

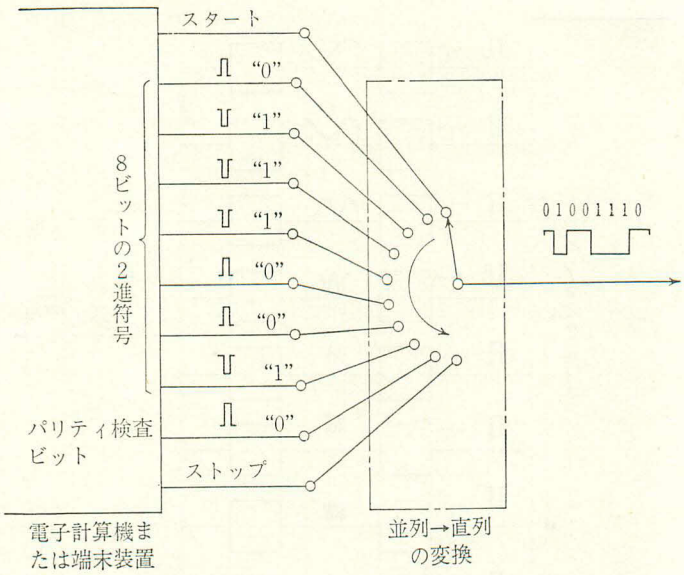


(a)　1ビットずつ移し換える場合



(b)　8ビット同時に移し換える場合

第 3.33 図　データの転送方法



第3.34図　並列から直列への変換

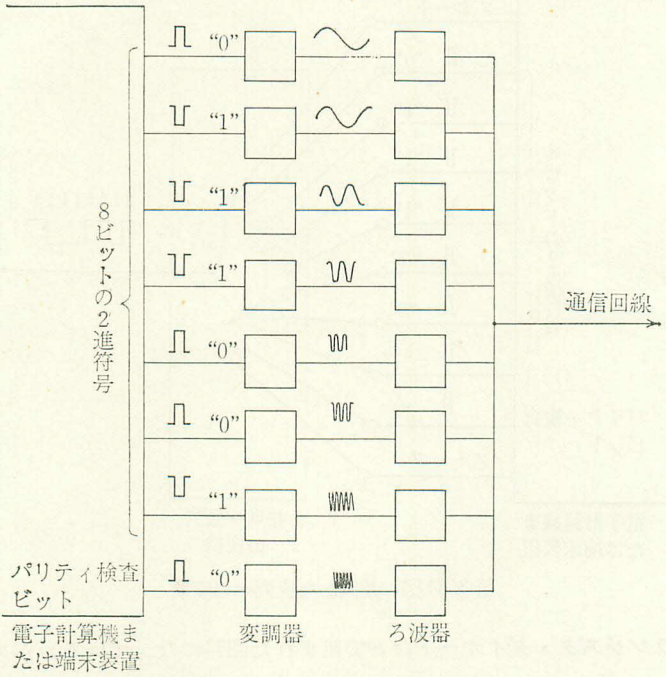
際にはトランジスタ・ダイオードなどで組まれた回路になっている）がスタートの位置からストップの位置まで順々に切り換えてゆき，それとともに1ビットずつデータを通信回線に送り出している。

並列伝送を行なうには，複数の通信回線を用いればよいが，**第 3.35 図**に示すように周波数の異なる複数の交流を用いることにより，1本の通信回線で並列伝送を行なうことができる。すなわち，この場合には，通信回線の周波数帯域を幾つかに分割して，その1つ1つに2進符号の各けたを割り当てることになる。図から明らかなように，並列伝送方式では並列→直列の変換は不要になるが，多数の変調器とろ波器が必要になり，直列伝送方式との優劣は一概には定められない。

3.3.2　通信速度

1.　通信速度の尺度　データ伝送においては通信速度，すなわち，1秒間に伝送しうるデータの量が重要な要素であり，その尺度にはデータ信号速度，データ伝送速度，変調速度の3種類がある。

データ信号速度はデータを伝送する瞬間速度であり，1秒当たりに伝送されるビ



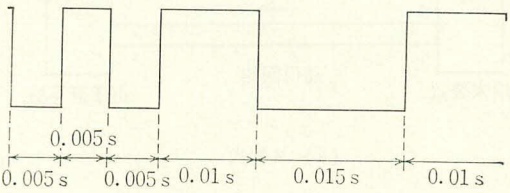
第3.35図　並　列　伝　送

ット数で表わされる。1秒間に 200 ビットを伝送する場合には 200 ビット/秒の速度であるという。

データ伝送速度は単位時間当たりの平均速度を表わし,その単位にはビット/秒,字/分などが使用される。一定の速さでデータが伝送されている場合には,データ信号速度とビット/秒で表わされたデータ伝送速度は一致する。

データ伝送は電圧の極性,周波数の高低,位相などにより "1", "0" を表わすのであるが,同じ電圧,周波数,位相が続いている最小の時間,すなわち,最短パルス時間長を秒単位で表わしたとき,この逆数を**変調速度**といい,**ボー**を単位として表わす。たとえば**第 3.36 図**では電圧がマイナスまたはプラスの状態をつづけている最小の時間は 0.005 秒であり,したがって変調速度は 1/0.005=200 ボーとなる。

2進符号を用いる直列伝送の場合には,データ伝送速度（あるいはデータ信号速度）と変調速度とは通常数値が一致するが,並列伝送の場合にはデータ伝送速度よ



第3.36図 変調速度

りも変調速度の方が小さくなる。たとえば第3.35図に示した並列伝送では，200ビット/秒の速さでデータを伝送するのに必要な変調速度は25ボーとなる。なお直列伝送の場合にも，第3.32図に示したような四相位相変調方式では，1回の位相変化で2ビットを送ることができるので，この場合の変調速度はデータ伝送速度の半分，すなわち，200ビット/秒のデータ伝送速度に対して100ボーの変調速度が対応する。

2. 周波数帯域と通信速度

周波数帯域の狭い通信回線を使ってデータ伝送を行なうと，信号波形のくずれ，すなわちひずみが大きくなり，信号を正しく受信できない。そして波形のくずれる程度は，通信速度が早いほど大きくなる。したがって，通信速度の早いデータ伝送を行なうにはそれだけ周波数帯域の広い通信回線を必要とする。

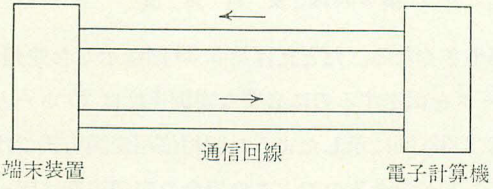
たとえば現在実用化されているデータ信号速度200ビット/秒のデータ伝送では，電話用の通信回線の周波数帯域300 Hz～3 400 Hzを6分割して使っているが，1 200ビット/秒の伝送ではこの300 Hz～3400 Hzの周波数帯域全部を使用する必要がある。

同じ周波数帯域を使っても，伝送方式をくふうすることにより，より高速度のデータ伝送を行なうことも可能である。たとえば四相位相変調方式を使えば300 Hz～3 400 Hzの周波数帯域の通信回線を用いてデータ信号速度2 400ビット/秒のデータ伝送が可能である。しかし，前にも述べたように，この場合には，通信回線の品質（信号対雑音比や位相周波数特性など）に対する要求が一層きびしくなる。

3.3.3 通信方式

1. 2線式と4線式　第3.37図に示すように，1対（2本）の線を使って送信

（a）2線式



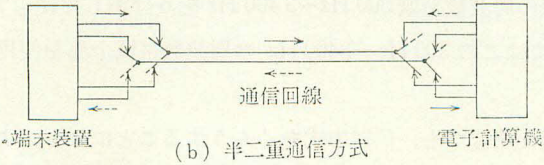
（b）4線式

第3.37図　2線式と4線式

と受信を行なう方式を**2線式**，送信方向と受信方向に各々1対（2本）の線を使用し，合計4線で送受信を行なう方式を**4線式**という。

2.　**単向通信方式**　　第3.38図（a）に示すように，データが常に一方向のみに

（a）単向通信方式



（b）半二重通信方式

（c）全二重通信方式

第3.38図　通　信　方　式

流れる方式を**単向通信方式**という。

3. **半二重**(1)**通信方式**　第3.38図 (b) に示すように，回線の両端において送受信を切り換えることにより，データの流れる方向を切り換えて両方向の通信を行なう方式を**半二重通信方式**という。この方式では，同時に両方向の通信を行なうことはできない。

4. **全二重**(2)**通信方式**　第3.38図 (c) に示すように，同時に両方向のデータ伝送が可能な方式で，単向通信方式や半二重通信方式が2線でよいのに対し，一般には4線式が使用される。

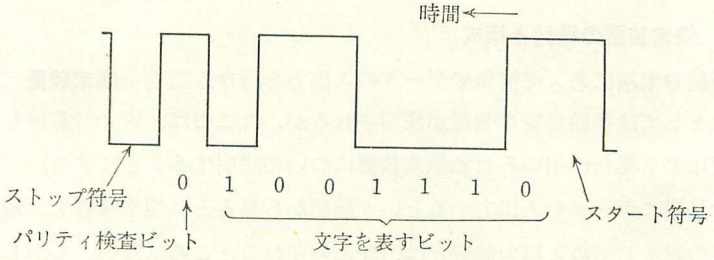
通信回線の周波数帯域がデータ伝送の通信速度に比して十分広い場合には，帯域幅を上下に2分して，それぞれ逆方向に使用して2線式で**全二重通信**を行なう場合もある。

3.3.4 伝送制御

データ伝送において，データを正しく送受信するためには種々の伝送制御が必要である。ここでは，そのうちのタイミング制御と誤り制御について説明する。

1. **タイミング制御**　データ伝送を行なうときには，パルス列のどこからどこまでが1つの文字を表わすかを明確にするために，送信側と受信側で同期を取る必要がある。このような制御を**タイミング制御**という。タイミング制御の方法にも各種あるが，**調歩式**(3)がもっとも広く用いられている。

調歩式は第 3.39 図に示すように，文字の初めにスタート符号を，また終わりにス



第3.39図　調歩式のタイミングの制御

(1) half duplex　(2) duplex　(3) start-stop system

トップ符号をつけるもので，受信側ではこのスタートストップ符号によって文字の区切りを判定する。

文字を7ビットで表わし，パリティ検査ビットを1ビット，スタートストップ符号をおのおの1ビットとすると，1文字は10ビットで構成されることになる。

2. **誤り制御**　データ伝送に際して，雑音や瞬断などの原因で符号が誤って受信されると，誤字を生ずる。この誤字の発生をできるだけ少なくする方法が**誤り制御**である。

誤り制御方式のうち，最も普通に行なわれているのは，第2章で述べたパリティ検査ビットを付加する方法である。パリティ検査ビットは，1文字中の“1”の合計が偶数（あるいは奇数）になるように“1”か“0”かに決められる。たとえば“1”の合計を偶数とする場合，1001110に対するパリティ検査ビットは“0”になる。受信側では“1”の合計を検査して，それが偶数（あるいは奇数）でない場合には誤りが発生したと判定する。パリティ検査のほか，**巡回符号（サイクリック・コード）**と呼ばれる特殊な符号をつけ加えて送信する検査方式もある。

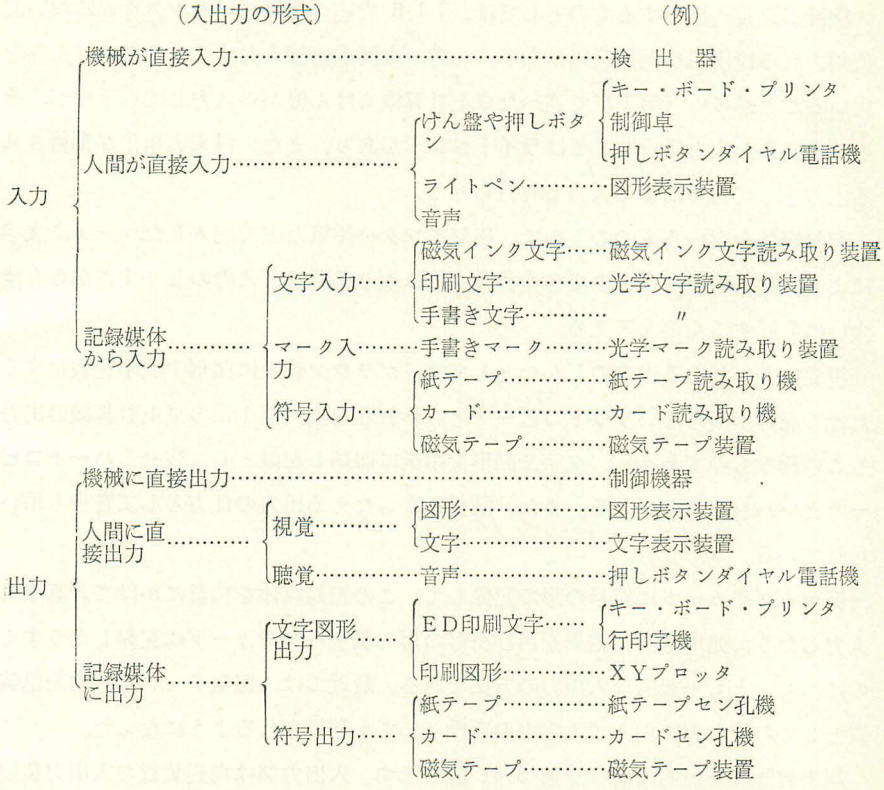
誤り制御方式にはそのほか，送信側で送信符号を記憶しておき，受信符号をそのまま送信側に返送して，両者を照合することにより誤りを見つける**返送照合方式**や，同じ符号を2回つづけて送信し，受信側で両者を照合する**遠送方式**などもある。

3.4 端末装置

3.4.1 端末装置の種類と構成

通信回線の末端にあって情報やデータの入出力を行なう装置を**端末装置**という。端末装置としては多種多様な装置が使用されるが，ここでは，電子計算機をオンラインで使用する場合に用いられる端末装置について説明することにする。

電子計算機にデータを入出力するという機能から見ると，端末装置と，電子計算機の周辺装置としての入出力装置とは，なんら変わるところはない。しかし，端末装置はあらゆる条件の場所に多数設置され，また，いろいろな人間に操作されることが多いために，性能よりも経済性と操作のしやすさに設計の重点が置かれている



第3.40図 端末装置の分類

こと，および必要に応じて，データ伝送を行なうための機能も付加されていることなどの点で周辺装置としての入出力装置と異なっている。

第2章のはじめに，代表的な入出力装置をデータの入出力の形式から分類して示したが，ここで第2章で説明しなかった入出力装置や，端末装置としてしか使用されないものも含めて分類しなおしてみると，**第 3.40 図**のようになる。例として挙げた装置には，検出器，押しボタンダイヤル電話機，制御機器のように端末装置としてしか使われないものがあり，また，制御卓のように周辺装置としてしか使われないものもあるが，大部分の装置は（性能の点での若干の違いはあるが）端末装置としても周辺装置としても使用される。

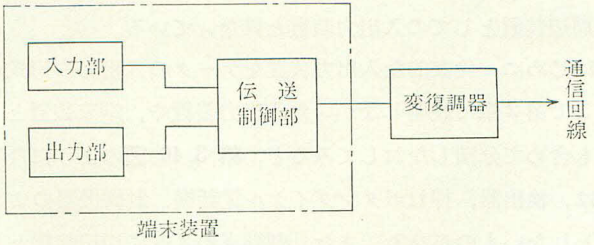
機械が直接入出力するものとしては，**3.1.6** で述べた制御用のシステムにおいて使用される検出器や制御機器がある。一方，人間が直接入力するものではけん盤を用いることが多い（第1章で述べた卓上計算機もけん盤から入力している）が，その他に特殊な入力手段としては**ライトペン**(1)があり，また，将来実用化が期待されるものとして音声による入力も考えられる。

記録媒体を用いるもののうちで，磁気インクや鉛筆などで記入したマークや文字により直接入力することのできる各種の読み取り装置は，入力のしやすさから今後大いに発展するものと考える。

視覚にうったえる出力のしかたとしては，ブラウン管上に図形や文字を表示するだけで記録が残らない**ソフトコピー**(2)といわれるもの（第1章の卓上計算機の出力もこの種のものである）と，文字や図形を用紙に印刷し記録として残せる**ハードコピー**(3)といわれるものがある。また，聴覚にうったえる出力の仕方として音声も用いられている。

紙テープやカードに符号の形で記録して，この記録媒体を装置にかけて計算機に入力したり，処理された結果を再び符号の形で紙テープやカードに記録したりするのは，もっとも一般的な入出力の方法である。最近では，磁気テープも補助記憶装置としての使い方に加えて入出力の手段としても利用されるようになった。

端末装置の一般的な構成を**第 3.41 図**に示す。入出力部は周辺装置の入出力装置と同様な機能を持っている。伝送制御部はタイミング制御や誤り制御などの伝送制御機能を有しており入力部，出力部および変復調器の動作を制御する。



第 3.41 図　端末装置の構成

(1)　light pen　　(2)　soft copy　　(3)　hard copy

端末装置によっては入力部，出力部のいずれか一方のみを持つものもある。また複数の入出力部を持ち，たとえばけん盤でも紙テープでも入力できるものもある。また，端末装置自体でその入力部からデータを投入すると出力部に出力できるような機能を持っているものがあり，たとえばけん盤をたたいて紙テープなどにせん孔し，この紙テープを用いてあらためてデータの入力を行なったりすることができる。

入力部，出力部，伝送制御部は一体となって1つの装置に組み込まれているものもあれば，おのおの独立した装置となっている場合もある。

3.4.2 キーボード・プリンタ(1)

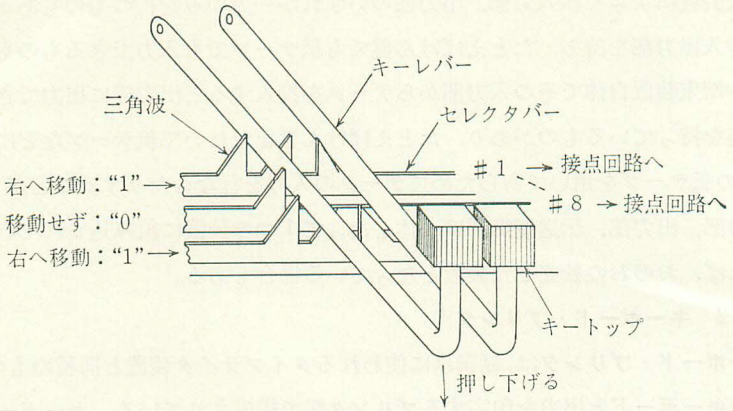
キーボード・プリンタは，制御卓に使われるタイプライタ装置と同種のもので，入力部のキーボードと出力を印字するプリンタ部で構成されている。キーボードとは数字，アルファベット，カナ文字，特殊記号などの文字を送信するためのキーを配列したけん盤のことをいう。またプリンタは，送られてきたデータにより文字や記号を印刷するものである。一般のキーボードプリンタでは，けん盤をたたいたとき，同時に該当する文字をプリンタ部で印刷する機能を有しているものが多い。

1. キーボード（けん盤部） 第1章において述べた卓上計算機においては，ボタンを押すことによってデータの入力が行なわれた。キーボード・プリンタの本質的な機能はこれと変わらないが，押し下げられたキーに相当する2進符号に変換するための回路は機械的に作られているのが普通である。この機械部分の構成を代表的な2例についてつぎに説明する。

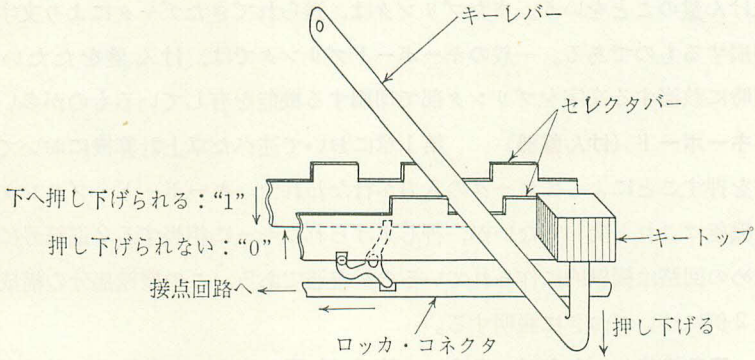
a. 三角波状セレクタバー方式 第 3.42 図に示すように符号，単位数だけのセレクタバーに三角波が並べられており，キー（キートップ）を押してキーレバーが下がると，三角波の斜面がキーレバーに押されて，各ビットに対応したセレクタバーが左右に移動し，セレクタバーに連なった接点回路により2進符号が作成される。

b. 方形歯状セレクタバー方式 第 3.43 図に示すように，各符号単位のセレクタバーが2本ずって構成されており，キーレバーが下がったときにセレクタ・バーの切り込みの有無によってセレクタバーに押し下げられるものと押し下げられ

(1) key boad printer



第3.42図　三角波状セレクタバー方式

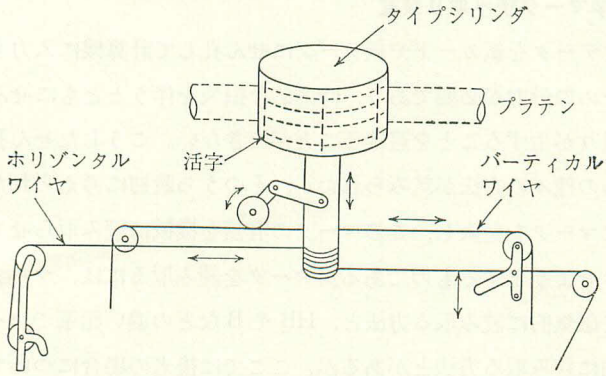


第3.43図　方形歯状形セレクタバー方式

ないものが生じ，その上下動がロッカコネクタの左右動に変換されて，2進符号が作成される。

2.　プリンタ（印字部）　　プリンタの印字方式には機械的なものと電子的なものがある。機械的なものは，従来から一般のタイプライタに使われてきたタイプバスケット式のほか，ゴルフボール状の球面に活字が作られているものを回転させながら印字するタイプボール式，タイプシリンダ式，タイプホイール式などの方式がある。ここでは最後の2者について説明する。

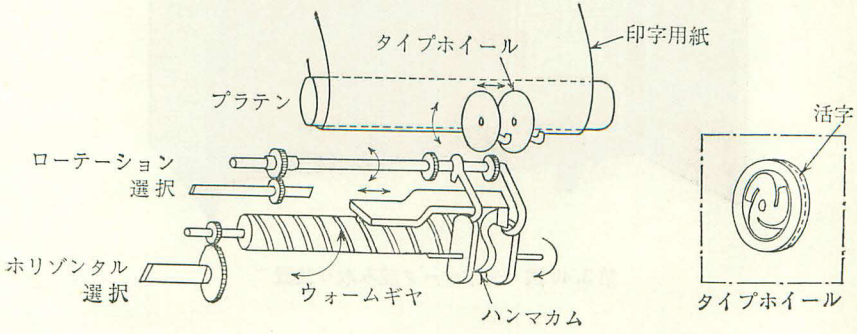
a.　タイプシリンダ式　　第 3.44 図に示すような円筒のタイプシリンダに活字



第3.44図　タイプシリンダ式

を円周方向に 32 字，縦方向に 4 ～ 6 字程度収容し，ホリゾンタルワイヤで高速に回転させ，バーティカルワイヤで上下動させて必要な活字の位置決めを行なう。印字はタイプハンマによって該当する活字をたたくか，またはタイプシリンダ自体をプラテンにたたきつけて行なう。プラテンは固定し，タイプシリンダが移動する。

b.　タイプホイール式　　**第3.45図**に示すように，円周上に 10 数個の活字を収容したタイプホイールを数個並べ，ホリゾンタル選択によってウォームギヤを動かし，タイプホイール全体を水平に移動させて該当するタイプホイールを所定の位置に固定する。ついでロテーション選択により該当する活字のところまでタイプホイールを回転させて印字する。

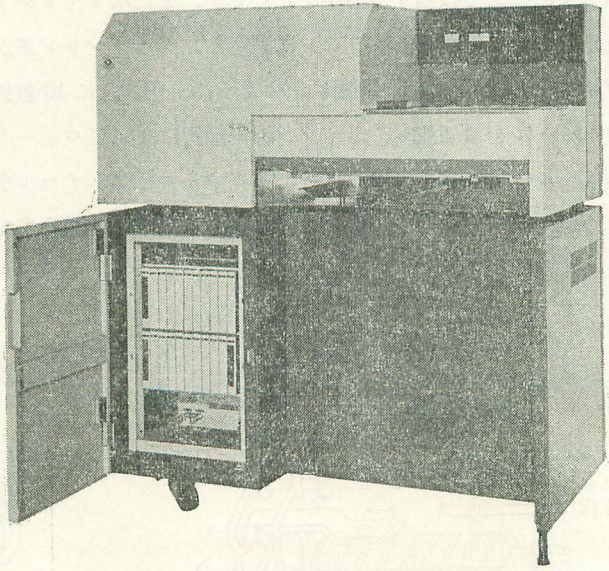


第3.45図　タイプホイール式

3.4.3　光学マーク読み取り装置(1)

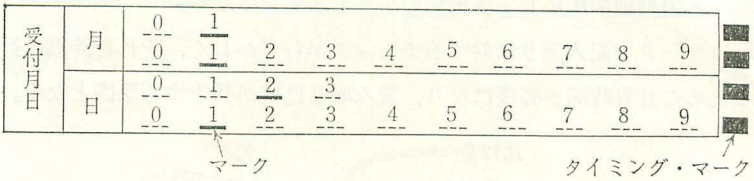
入力すべきデータを紙カードや紙テープにせん孔して計算機に入力する方法は，せん孔するための労力が必要であり，時間的な損失を伴うとともにせん孔に際してある程度の誤りが生ずることを避けることができない。こうしたせん孔という過程をなくすための種々の方法が試みられたが，そのうち最初に考えられた方法は，カードや帳票にマークを記入し，このマークの有無を機械に読み取らせてデータを計算機に入力させようとするものである。マークを読み取るには，マークを磁気インクで記入して磁気的に読み取る方法と，HB やＢなどの濃い鉛筆でマークを記入しこれを光学的に読み取る方法とがあるが，ここでは後者の場合について説明する。

第 3.46 図は光学マーク読み取り装置の外観図である。マークを記入する用紙を**マーク・シート**という。このシートには，入力すべきデータの項目と，データの内



第3.46図　光学マーク読み取り装置

(1)　optical mark reader：略してOMR



第3.47図 マーク・シートの記入例

容を表わすための数字が記入されている。たとえば第 3.47 図に示すように受付月日の項目があり，月を表わすための０～１と０～９の２列の数字，および日を表わすための１～３と０～９の２列の数字が並べられている。いま，11月21日という受付月日を入力したい場合には，図に見るように第１列の１の下と第２列の１の下および第３列の２の下と第４列の１の下の定められた個所にマークを記入する。

このようなマーク入力方式の利点は，

（1） 記入方法が単純で，記入者に特別の技能が要求されない。

（2） １個のマークに固有の意味を持たせることができるので，紙テープやカードの場合と異なり部課名，品名，地名などを符号化して記録する必要がない。

（3） 用途ごとにかなり自由な見やすい様式の用紙を使用できる。

（4） せん孔の手間が不用なので，入力データ作成の費用が低廉になる。

（5） 記入誤りを人間の目で容易にチェックできる。

（6） 情報の発生者が直接入力データを作成しうるので，せん孔の場合のように途中で介在する人間の勘違いによる誤りを防止できる。

一方この方式にもつぎのような欠点がある。

（1） 記入誤りが比較的多いこと。

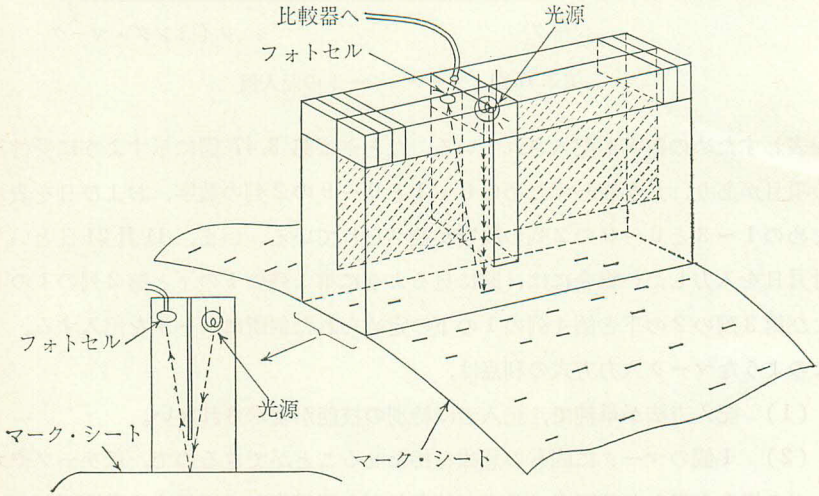
（2） マークの濃淡や，マークの消し残し，しみ，あるいは汚れなどの原因にもとづく誤読が発生しやすい。

（3） マークの位置を基準に読み取りが行なわれるので，マーク・シートの印刷には寸法精度の高いものが必要である。

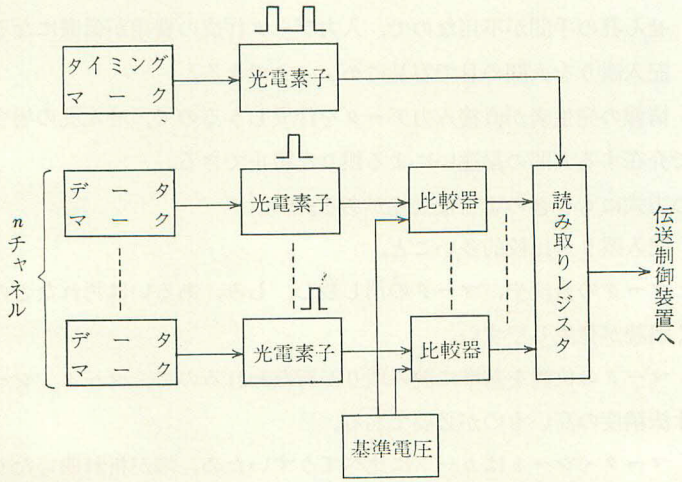
（4） マーク・シートはカードに比べてうすいため，端が折れ曲ったり，紙送りの不良を起こしたりする故障が多く，また，湿度により寸法が変わりやすい。

（5）　入力の種類に応じて多種類のシートが必要になる。

（6）　マークの記入誤りに対するチェックがむずかしく，それを詳細に行なうとそのための計算時間が必要になり，読み取り速度が低下する原因となる。

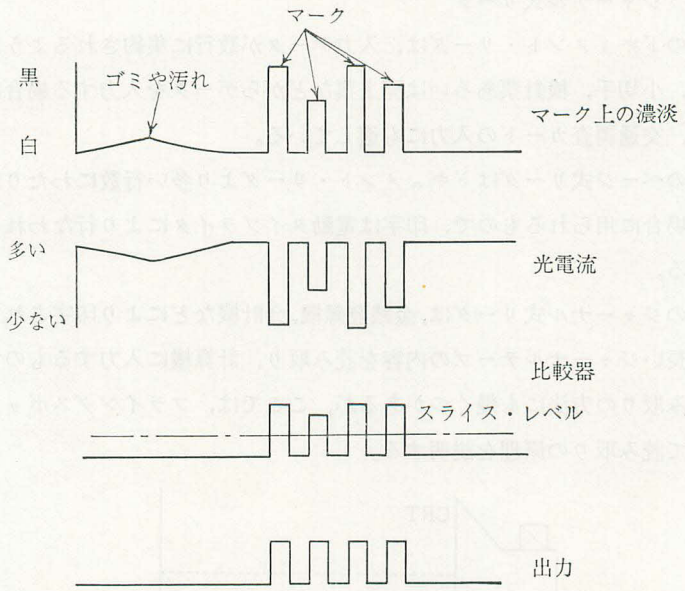


第3.48図　読み取りヘッド原理図



第3.49図　読み取り部ブロック図

第 3.47 図に示すように，マーク・シートの右側にはタイミング・マークが印されており，このタイミング・マークを基準にして，記入されたマークの有無の検出を行なうことにより，紙送りのむらや紙の伸縮の影響を軽減することができる。マークの読み取りは，第 3.48 図に示すように，マークの有無による反射光量の変化をフォトセルで検出して行なう。同一行にある複数のマークを同時に検出するため，検出装置はマークの記入位置の数だけ横に並べられている。フォトセルで検出されたマーク信号は，第 3.49 図に示すように基準電圧と比較され，第 3.50 図に示すように 2 進符号に変換されて読み取りレジスタに記憶される。



第 3.50 図

3.4.4 光学文字読み取り装置

光学文字読み取り装置は，紙の上に印字あるいは手書きされた文字を直接読み取り，計算機の入力とする装置である。

手書きの文字を直接読み取ることができれば，データの入力はきわめて簡易化されるが，文字を読み取る機能は現在まだまだ幼稚であり，線の太さと濃さをそろえ

一定の大きさと形に文字を記入しないと読み取りが困難となるため，文字を書くのに非常な困難が伴う状態である。また使用しうる文字の種類も限られているなど，文字読み取り装置の適用範囲は現在かなり限定されているが，今後オンライン・システムの普及，発展に伴って，広く利用されるようになるものと考えられる。

光学文字読み取り装置は，使用される用紙の種別によって3つに分けることができる。

(1)　ドキュメント・リーダ

(2)　ページ式リーダ

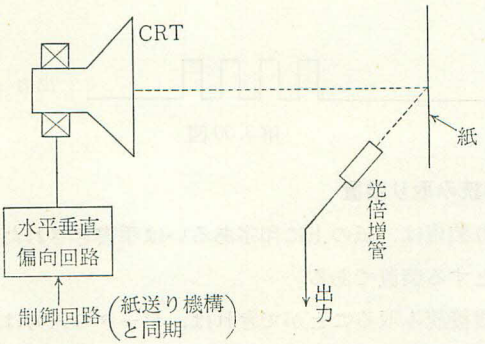
(3)　ジャーナル式リーダ

(1)のドキュメント・リーダは，入力データが数行に集約されるような簡単なもので，小切手，検針票あるいは売上票などからデータを入力する場合によく用いられ，交通調査カードの入力にも適している。

(2)のページ式リーダはドキュメント・リーダより多い行数にわたり印字されている場合に用られるもので，印字は電動タイプライタにより行なわれるのが普通である。

(3)のジャーナル式リーダは，金銭登録機，会計機などにより印字されてできてくる細長いジャーナルテープの内容を読み取り，計算機に入力するものである。

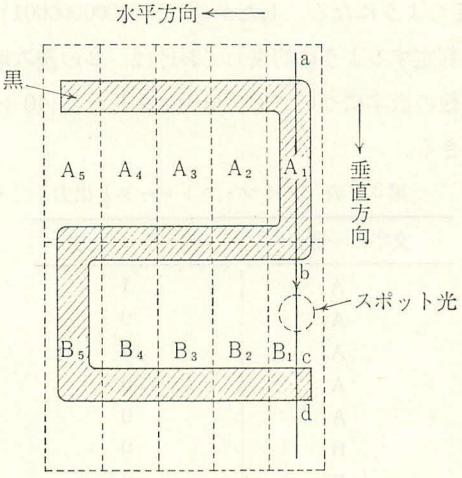
文字読み取りの方法にも幾くつかあるが，ここでは，フライングスポット方式を例にとって読み取りの原理を説明する。



第3.51図　文字読み取りヘッドの原理

本方式の読み取りヘッドは**第 3.51 図**に示すように構成されており，ブラウン管，すなわち陰極線管(CRT)[1]からのスポット光が紙上に書かれた文字に当てられている。このスポットは，外部からの制御によって紙上を走査できるようになっている。

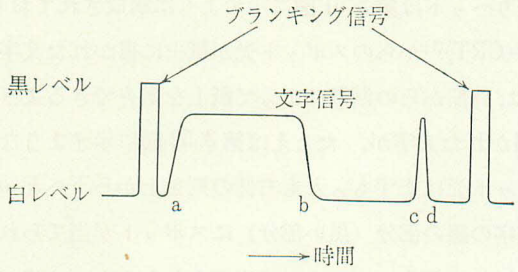
いま，紙上に書かれた文字が，たとえば**第 3.52 図**に示すような数字の 2 であったとしよう。スポット光はまずもっとも右側の列を上から下へと a，b，c，d の順に動いていく。文字の線の部分（黒い部分）にスポットが当てられたときには，光が反射せず，それ以外の部分にスポットが当てられたときには光が反射するので，この反射光を光倍増管に受けて電気信号に変換し増幅すると，**第 3.53 図**のような波形が得られる。



第 3.52 図 光 学 文 字

ここで，第 3.52 図に示すように，2 という文字を $A_1 \sim A_5$，$B_1 \sim B_5$ の 10 区画に分割する。そうしてこの区画の縦の長さ，すなわち，文字の高さの半分（これをメイン・ストロークという）だけ黒い部分が続いたときには，この区画を“1”とし，そうでない場合は“0”で表わすものとする。このためには第 3.53 図の波形で a b 間のように黒レベルがある幅だけ続いたときにだけ出力が現われ，c d 間のように瞬間的に黒レベルになった場合は出力が現われないようにしてやればよい（ここで

(1) cathod ray tube: 略してCRT



第3.53図　光倍増管の出力

は出力が出たときを“1”に，出ないときに“0”を対応させている)。

このようにして文字2を A_1, B_1, A_2, B_2, ……, A_5, B_5 の順で走査するとき，その出力は**第3.2表**のようになる。したがって，「1000000001」の出力が得られたときは，これを2と判定するように約束しておけば，2の読み取りを行なうことができる。同様にして他の数字についても，出力に表われる10けたの2進符号により判別することができる。

第3.2表　メイン・ストロークと出力

文字の一部分	出　力
A_1	1
A_2	0
A_3	0
A_4	0
A_5	0
B_1	0
B_2	0
B_3	0
B_4	0
B_5	1

このように，数字だけの読み取りは比較的簡単な走査を行なうことにより判定できるが，さらに一般的に文字を読み取る場合は，水平，垂直，ななめなどの走査を行なうことによって，文字の特徴が符号として現われるようにする必要がある。こうして文字の識別をもっとも有効に行なえるような走査方法，すなわち，特徴検出機能の良否が文字読み取り装置を開発する上での重要な課題となっており，内外に

おいて研究が進められている。日本では，片かな，平がな，さらには漢字があるために，その判別は一層困難である。そのうえ手書きの場合には，濃淡も形状も一定してないため，識別はさらに困難になる。現在の文字読み取り装置の読み取り速度は通常 300～400 字/秒程度であるが，1000 字/秒のものも開発されつつある。また読み取り誤り率は，数字のみの場合 10^{-5}～10^{-6} 程度の値が得られている。

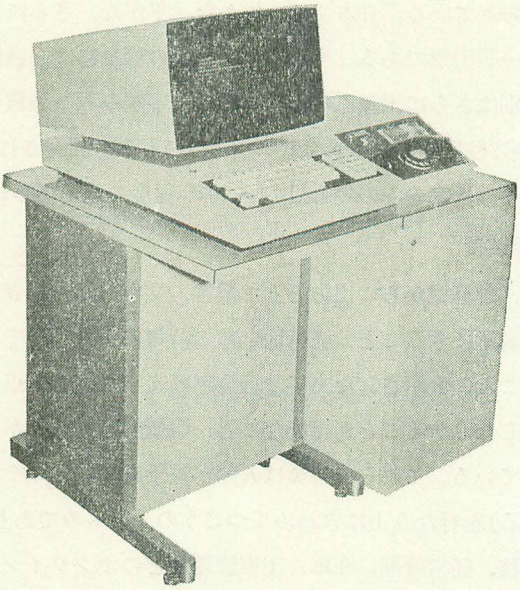
3.4.5 文字表示装置

文字表示装置は，計算機からの出力として数字，文字，記号を表示する装置でありデータを高速で表示できること，表示面全体の情報により判断できること，編集機能を持っていること，表示された文字や行全部の文字などを修正したり変更したり，さらには消去したりすることが可能なこと，機械的可動部分がなく静かなことなどの特徴を持っている。これらの特徴は人間が電子計算機とのやりとり（会話という言葉で呼ばれる）を行なう上にきわめてつごうのよいものであり，教育，医療，情報検索，預金業務，経営管理，在庫，工程管理などのオンライン・リアルタイム・システムの端末装置として広く使用されるようになってきた。文字表示装置の外観を**第 3.54 図**に示す。文字表示装置は，**第 3.55 図**に示すようにデータ・バッファ記憶部，文字発生部，表示部およびけん盤から構成されている。

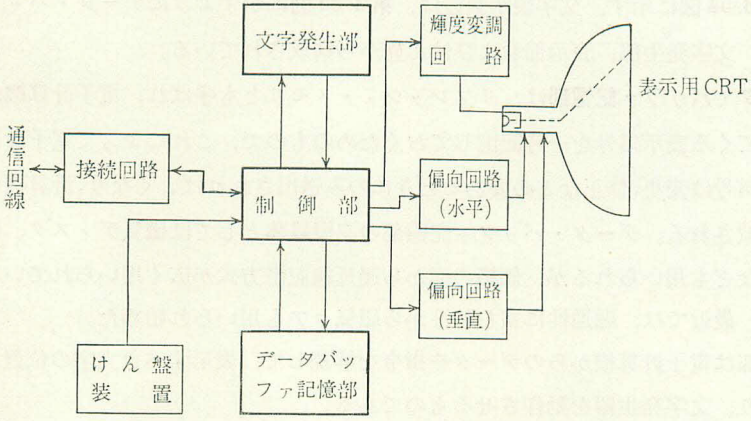
データ・バッファ記憶部は，リフレッシュ・メモリとも呼ばれ，電子計算機から送られてくる表示信号を一時記憶しておくためのもので，これによって電子計算機からの信号は変更，修正など必要あるときにのみ送出されればよくなり，計算機の負担が軽減される。データ・バッファ記憶部の記憶媒体としては磁気ディスク，磁気ドラムなども用いられるが，価格の面から遅延線記憶方式が広く用いられている。しかし，最近では，融通性に富む特性から磁気コアも用いられ始めた。

制御部は電子計算機からのデータや指令を解読して，表示すべき文字の位置を決定したり，文字発生器を動作させるものである。

文字発生部は，制御部の指令にしたがって，記憶部から読み出された文字符号を文字表示するための映像信号に変換し表示部に送出するものである。表示しうる文字の種類は，数字，アルファベット，特殊記号などのほか，カナ文字や漢字までを表示できるものもある。



第3.54図　文字表示装置



第3.55図　文字表示装置の基本構成

電子計算機からの出力を表示する表示部は，現在 CRT が広く用いられている。その他エレクトロ・ルミネッセンス(1) (EL)，プラズマ・ディスプレイ・パネル(2)

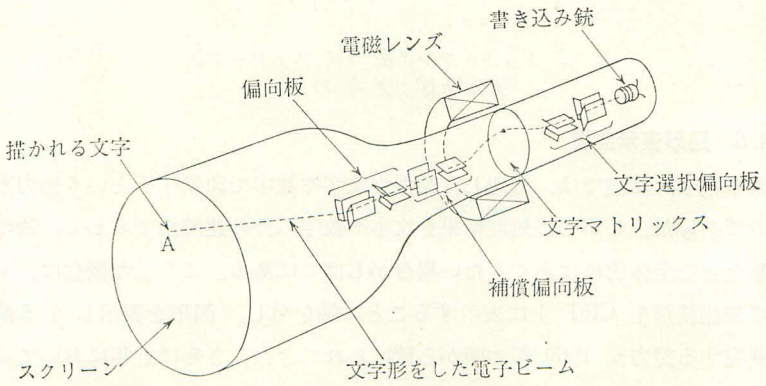
(1)　electro luminescence：略してEL　　(2)　plusma display panell：略してPDP

(PDP) あるいは光の散乱を利用した液晶などが現在開発中であり，将来の表示手段として期待されている。

文字表示装置には，操作員（オペレータ）により表示内容の修正，変更および電子計算機へのデータの送信のため，通常けん盤が付加されている。けん盤のキーを押すと，それに対応する符号が記憶部にはいり，表示内容が変更される。変更しようとする文字の位置を指定するには，画面上に現われるポインタ（通常カーソルと呼ばれる）を特定のキーで移動させて行なう。また，直接ライトペンで文字位置を指示する方法もある。

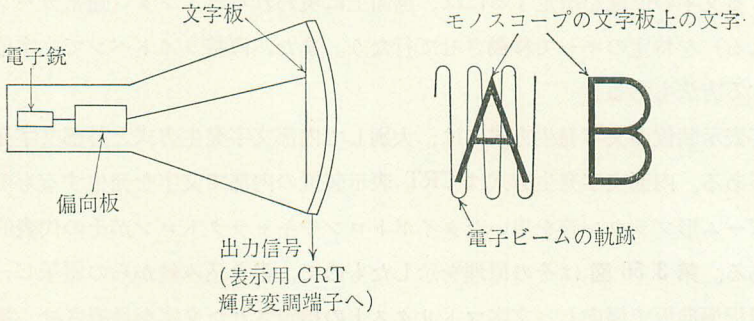
文字表示装置の文字発生方式には，大別して内部文字発生方式と外部文字発生方式とがある。内部文字発生方式は CRT 表示装置の内部で文字を発生するもので，成形ビーム形ブラウン管を用いたタイポトロンやキャラクトロンがその代表的なものである。**第 3.56 図** はその原理を示したもので，書き込み銃からの電子ビームを文字選択偏向板で偏向し，文字マトリクス上の指定された文字を通過させ，電磁レンズで集束した後再び偏向板により偏向して，ブラウン管の所要の位置に照射し文字を描かせるものである。

外部文字発生方式は文字を描くための映像信号を別に発生して通常のブラウン管に加えるもので，**第 3.57 図** に示すような CRT モノスコープを用いるモノスコー

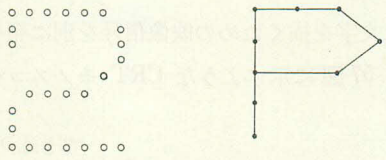


第 3.56 図　内部文字発生方式の概略図（文字マトリクスには普通 64 種の英数字，記号がある）

プ法，**第3.58図**に示すように文字や記号を点のら列で表わすドット・アレイ法および文字や記号を線分をつないで折線で表現するストローク法などがあり，またこれらの変形としてスターバースト法やフライングスポット法（モノスコープ法と原理は同じであるが，モノスコープの代わりに，フライング・スポット・チューブを用いるもの）なども用いられている。



(a) モノスコープの構造　(b) 電子ビームの軌跡
第3.57図　モノスコープ法



(a) ドット・アレイ法　(b) ストローク法
第3.58図　文字の表示

3.4.6　図形表示装置

従来の電子計算機では，処理した結果を文字や数字で印刷するという出力形式が大部分であった。しかし，処理結果を文字や数字以外の連続量で表わし，物の形状や波形などを全体的にはあくしたい場合がしばしばある。こうした概念は，レーダなどの観測波形を CRT 上に表示することに端を発し，図形を表示しうる端末装置を開発する努力が 1960 年中頃から続けられてきた。さらに近年においては，二次元平面的な図形のみでなく，三次元の図形を表示する試みも現われている。図形表示装置は，現在まだ特殊の領域でしか利用されていないが，今後は図形処理用の

ハードウェア，ソフトウェアの開発とあいまって，広く利用されるようになるものと期待される。

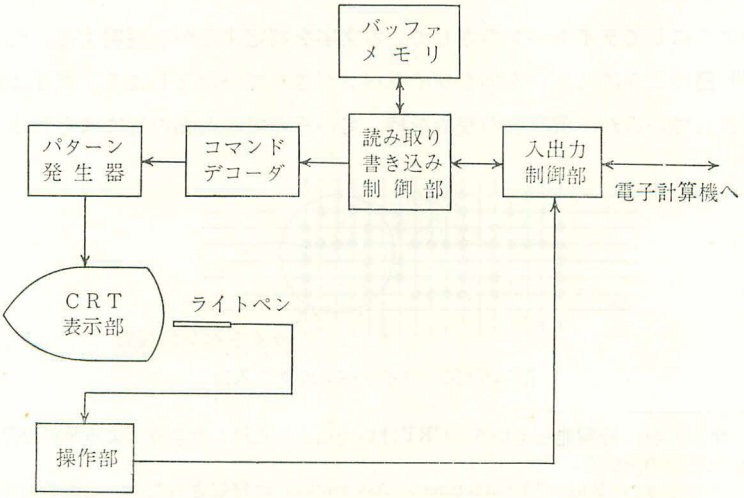
図形表示装置の出力装置としては CRT を用いた表示装置が使用され，また，入力装置としては各種のキーのほかライトペンが使用されている。

1. CRT 表示装置（出力装置）

第 3.59 図に図形表示装置のブロック図を示す。電子計算機から，表示される図形に応じて**コマンド**(1)という一種の命令が送られてくる。これはバッファメモリ（記憶部）に記憶される。このコマンドの形式は通常の電子計算機の命令と同様の構成をとっている。コマンドは，表示装置内のコマンド解読器によって実行すべき内容が解読される。コマンドの命令内容が点，文字，直線，円のいずれであるかが解読されると，これによって，点発生器，文字発生器，直線発生器，円発生器のいずれかが駆動され CRT 上に該当する図形を表示する。

一般に，表示された画面は 30～60 回/秒の周期で繰り返し再生される。

CRT の画面は，一般に縦横 2 進 10 ビットすなわち 1,024 個ずつの点格子で座



第 3.59 図　図形表示装置のブロック図

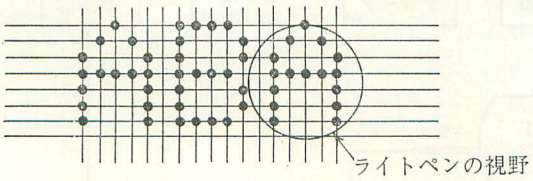
(1)　command

標付けされており，これら約 100 万個の点が電子ビームの照射を受けて，それぞれ光ったり消えたりすることにより，文字や図形を表示している。

テレビジョンのような広角ブラウン管では，加える電圧レベルと電子ビームのふれの直線性が悪くなり，**解像度**(1) も低くなるが，CRT 表示装置では，テレビの受像管より狭角で，直線性がよく解像度も 2 倍程度のものが用いられている。使用するけい光体については，ライトペンからは残光性の短いことが要求され，一方，人間からは，ちらつきのない残光性のあるけい光体が要求されるが，P 7 (2) という青白色のけい光と黄緑色の残光を持つけい光体を用いることによってこれを解決している。

2.　ライトペン(3)**（入力装置）**　図形表示装置を操作しデータを入力するには，ファンクションキー，ライトペンあるいは文字キー・ボード，さらにコマンド用のキーや表示位置を決めるキー等によって行なうが，ここではライトペンについてのみ説明する。ライトペンは，ペン状の棒の先端に小孔があり，その内部に CRT 上の青白色のけい光を検出するための光検出器を含む構造を有し，図形のある 1 点，あるいは文字を指定したり，図形に直線や曲線を加えたり，図形を移動させる機能を持っている。

どのようにしてライトペンのさしている文字を判定するかを説明する。たとえば**第 3.60 図**のようにAという字をライトペンがさしていたとしよう。画面は繰り返し走査されているが，黄緑色の残光を持っているので，人間の目にはそれほどちら



第 3.60 図　ライトペンによる入力

(1)　resolution：**分解能**ともいう。CRT けい光面上の近接した 2 点による像が分離して現われる能力をいう．

(2)　アメリカの Radio Manufacturer's Association に登録されたけい光面の番号であり P 1 から P 34 までである。P 7 は，硫化亜鉛と銀，および硫化亜鉛と銅の混合けい光体を塗布したもの

(3)　light pen

つきは感じない。しかし，青白色の残光はたかだか数 10μs であるので，青白色にのみ感知するライトペンは，ペンがいまさしているAという文字が走査されている瞬間だけ出力を生じる。こうして，走査が開始されてからライトペンが光を検知するまでの時間を計れば，ライトペンの位置が算出でき，したがって，ライトペンがAをさしていることが判明する。

このデータを電子計算機に送って，ライトペンの指示位置に対応した処理を行なわせることもできる。また，ライトペンの位置のずれを前と同様の原理で検出することによって，ライトペンの動きを検出することもできる。こうして，ライトペンで画面に図形を書くことにより，いわば図形のデータを電子計算機に入力することができることになる。

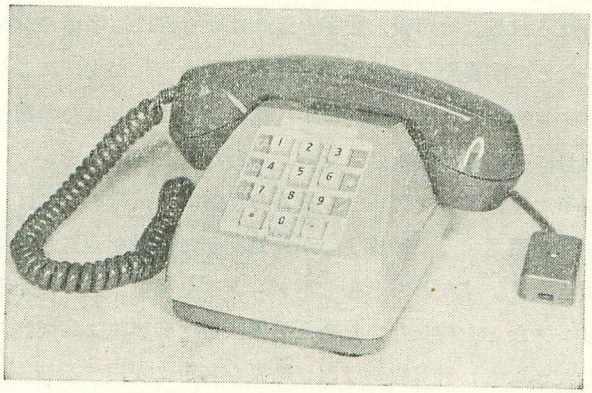
3.4.7 押しボタンダイヤル電話機

近年，サービスが始められた**押しボタンダイヤル電話機**は，逐次全国的にサービス範囲が拡大され，従来の回転ダイヤル電話機に代わって用いられるようになると考えられるが，この押しボタンダイヤル電話機は，つぎの特徴を有する。

（1） 相手側と接続が完了した後においても，信号を伝送することができる。

（2） 数字ボタンのほかに2個の機能ボタンを有し，これと数字ボタンとの組み合わせによって各種の符号を作ることができる。

これらの特徴を利用することにより，押しボタンダイヤル電話機をオンライン・



第3.61図 押しボタンダイヤル電話機

第3.3表　押しボタンダイヤル電話機の数字配列と周波数

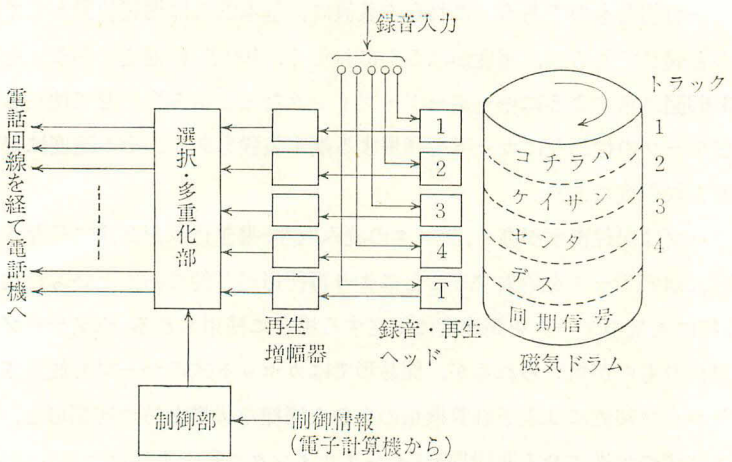
低群〔Hz〕＼高群〔Hz〕	1209 (H_1)	1336 (H_2)	1477 (H_3)
697 (L_1)	1	2	3
770 (L_2)	4	5	6
852 (L_3)	7	8	9
941 (L_4)	●	0	⊙

〔注〕 1 を押すと 697 Hz と 1209 Hz の周波数の交流が送られる。

システムの端末機として利用することができる。**第3.61図**にその外観を示す。

押しボタンダイヤル電話機は，**第3.3表**に示すように低群4波，高群3波の7個の発振器を有しており，1つのボタンを押すと低群から1波，高群から1波を選んで送出されるようになっている。この2波を選び出す組み合わせの種類は，4×3＝12 通りあるので，0～9の数字と2個の機能ボタン（赤ボタンと青ボタン）を表わす信号を作ることができる。押しボタンダイヤル電話機を用いて電話をかける場合には，相手電話番号に応じて0～9の数字ボタンを押すと，2周波の組み合わせ信号が順次送出される。電話局側ではこの信号を多周波受信機で受信し，どの数字が送られてきたかを判別して，あとは回転ダイヤル電話機と同じ方法で交換機が交換接続を行なう。同様な方法で押しボタンダイヤルより電子計算機のセンタに割り当てられた電話番号を送出すると，押しボタンダイヤル電話機が電話回線を通して電子計算機と接続される。その後，数字を送りたいときは数字ボタンを押し，また＋，－，×，÷などの演算記号を送りたいときは定められた規則にしたがって機能ボタンと数字ボタンを続けて押すと，電子計算機側では，これを多周波受信機で受信し，数字や記号に変換して要求された演算を行なうことになる。

押しボタンダイヤル電話機をデータの投入の目的に用いる場合には，電子計算機側は多周波受信器を用意するだけでよいが，処理した結果を返送する場合は，特殊なくふうが必要となる。すなわち，押しボタンダイヤル電話機は，処理結果を表示するためのプリンタなどを持たないため，結果は必ず音声によって送り返さなければならない。したがって，計算機側は処理結果を音声に変換する音声応答装置が必要となる。



第3.62図　原理図（録音編集方式）

音声応答装置は，電子計算機のセンタに置かれる装置であるが，押しボタンダイヤル電話機との関連が深いので，ここで簡単に説明しておく。

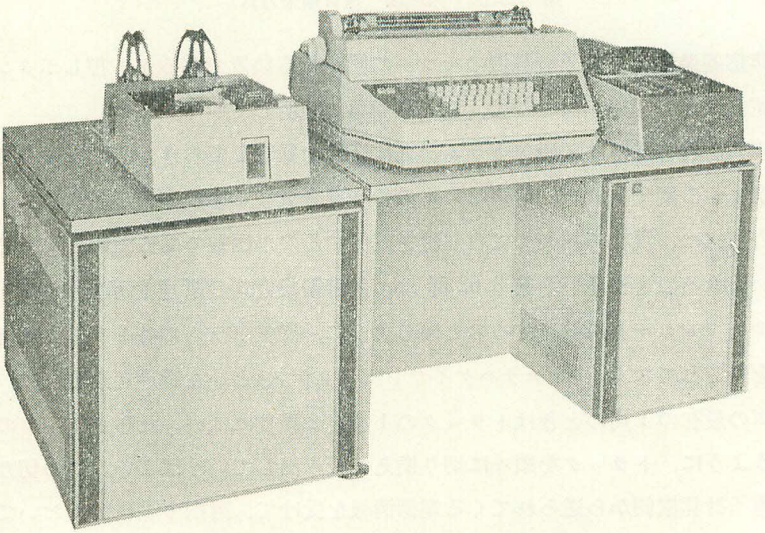
音声の応答方式には，磁気ドラムなどに音節を録音しておき，それを組み合わせて文章にする編集方式と，音声そのものを音声合成する合成方式がある。しかし，後者の方式は，現在の技術では自然性を確保するのが困難であるため，前者の方式が用いられることが多い。**第3.62図**は，録音編集方式の原理を示したものである。テープレコーダに録音するのと同じように，磁気ドラムの各トラックに音節や単音を録音しておく。「コチラハケイサンセンタデス」という音声を送るには，磁気ドラムの最初の1周のときはトラックの1を，2周目にはいったらトラックの2をというように，トラックを順々に切り換えて読み出していけばよい。この切り換えは，電子計算機側から送られてくる制御情報を受けて，選択多重化部において行なわれる。またこの選択多重化部は，1台の装置で多数の端末装置に音声を送り出すための多重化の機能も果している。なお磁気ドラムは，音声を正しい位置から読み出すことができるように音節のはじめと終わりを示す同期信号も記録されている。

3.4.8　紙テープ，紙カード，磁気テープを使用する端末装置

紙テープ読み取り機と紙テープせん孔機は，データ伝送用の端末装置として現在

もっとも一般的なものである。これらの装置は，基本的には周辺装置として用いられるものと同じであるが，速度がはるかにおそく，100字/秒程度である。そして通常は**第3.63図**に示すようにキーボード・プリンタなどと組み合わせて使用される。また，紙テープの代わりにカードを利用する端末装置もあり，その速度は毎分数十枚から数百枚程度である。

磁気テープは記録密度が高く，データの読み取り，書き込みを高速に行なうことができ，また，前のデータを消去することにより再使用が可能であるといった利点があるため，特に大量のデータを高速度で伝送する場合に使用される。磁気テープには電子計算機用のものが用いられるが，簡易形ではカセット式のテープも使用されている。磁気テープ装置には電子計算機用のものと同様な機構を持つ同期形と，磁気テープを1字ずつ歩進させる非同期形(インクリメンタル形とも呼ばれる)とがある。



第3.63図　キー・ボード・プリンタと紙テープ読み取りせん孔機を持つ端末装置

3.5 摘　要

1. データをその発生地点から電子計算機に投入したり，また，処理した結果をそれを必

要とする地点に出力する過程において，人手の介在を必要とするものをオフラインといい，人手の介在を必要としないものをオンラインという。

2. 外部事象の発生に応じて，限られたある時間内にデータを処理する方式をリアルタイム処理といい，そうでないものをディレィドタイムという。

3. オンライン・リアルタイム処理を行なうシステムをオンライン・リアルタイム・システムといい，電子計算機，通信回路，端末装置から構成される。

4. 電子計算機を利用したシステムには，一括処理システム，メッセージ（ある意味を表わすデータのひとかたまり）の交換を行なうシステム，端末装置から送られてきたデータをリアルタイム処理して送り返すシステム，多くの利用者が共同で独立して電子計算機を利用できるようなシステム，すなわち，タイムシェアリングシステムなどがある。

5. 複数のプログラムから適当に選び出しながら処理をすすめていくことにより，みかけ上同時に処理されているようにした多重プログラミングなど，多くの機能が高性能の電子計算機に要求されている。

6. あるプログラムの実行中に割り込んで別のプログラムを実行することを割り込みといい，多重プログラミングの場合のほか機械の誤動作やプログラムの誤りなどの場合にも使われる。

7. 複数のプログラムが主記憶装置を使用する場合には，主記憶装置の割り当てが必要である。プログラミングに際して，アドレスは相対的に付された相対アドレスで書かれているが，実行の際には，そのときの主記憶装置の使用状況によって装置個有のアドレス，すなわち絶対アドレスに変換される。

8. オペレーティング・システムは，操作員を助けたり，電子計算機の能力を最大限に生かすためのプログラムの集りで，管理プログラム，支援プログラム，応用プログラムから構成される。

9. プログラム誤りによって他のプログラムで使用している領域を侵害しないようにする機能として，記憶保護の機能がある。

10. オンライン・システムでは通信回線を通して端末装置を制御する機能として，通信制御の機能が要求される。一般にはこの機能を果たすために通信制御装置が用いられている。

11. システムの信頼性や処理効率の向上の目的で，複数の電子計算機を組にした電子計算機複合体が用いられることがある。

12. データ伝送の方式には直流伝送と交流伝送があり，交流伝送には振幅変調，周波数変調，位相変調などの変調方式がある。また，1ビットずつ伝送する直列伝送と，何ビットかを同時に伝送する並列伝送がある。

13. 通信速度にはデータ信号速度，データ伝送速度，変調速度の3種類の尺度がある。一般に通信速度が早いほど，広い周波数帯域の通信回線を必要とする。

14. 送信と受信を2本の線で行なうか，4本の線で別々に行なうかによって，2線式と4線式の区別がある。また，データの流れ方により単向通信方式，半二重通信方式，全二重通

信方式の区別がある。

15. データを正しく送受信するため，タイミング制御や誤り制御などの伝送制御が行なわれる。

16. キーボード・プリンタはデータを入力するキーボード（けん盤）と印字を行なうプリンタ部からなる端末装置である。

17. 光学マーク読み取り装置は鉛筆などで書かれたマークを読みとる装置である。

18. 光学文字読み取り装置は印刷された文字または手書きの文字を読み取る装置である。

19. 文字表示装置は CRT などに文字を表示する装置である。

20. 図形表示装置は出力部として CRT 上に文字のほか各種の図形を表示できる装置であり，入力部としてライトペンなどが使用される。

21. 押しボタンダイヤル電話機は従来の回転ダイヤル電話機にない特徴を有しており，オンライン・システムの端末機としても使用することができる。

3.6 練 習 問 題

1. データ信号速度 50 ビット/秒，200ビット/秒，1200ビット/秒の通信回線を使って，100 字からなるデータを連続して送りたい。ただし，制御用の符号は考えないものとし，また1字はパリティ検査ビットを入れて8ビットで構成され，調歩同期用のスタートストップのパルスは他のビットのパルス幅と同一であるとする。伝送制御などの時間を除き全文字を送るのにどのくらい時間がかかるか。

2. 正しい文章に○をつけよ。

（1） タイムシエアリング・システムは，大量のデータを入出力するのに適している。

（2） 電子計算機に問題を与えてから解答す出るまでの時間をスループットという。

（3） 文字の形でデータを出力するのは簡単であるが，文字の形で入力するのは技術的にむずかしい。

（4） デュプレックス・システムでは，予備側の電子計算機は常時仕事をさせずにあけておかねばならない。

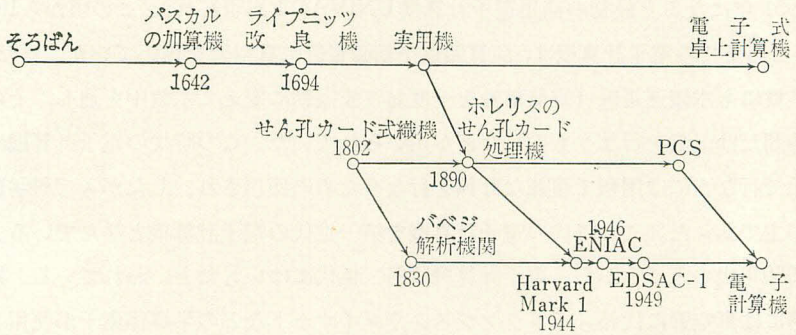
（5） オフライン処理におけるターン・アラウンド・タイム（データを入力してから処理された結果が出力されるまでの時間）は，オンライン・リアルタイム処理の場合に比べて長い。

3. 入出力装置には，動作不動作の2種類の状態があり，その状態を調べる命令があったとする。割り込み回路がないときに，入出力装置の動作終了にもとづく割り込みと同様な機能を果すためには，どのようなプログラムにしておけばよいか。また，この場合どのような問題があるか。第3.12図を例にとって説明せよ。

第4章　電子計算機の歴史と今後の動向

4.1　電子計算機の歴史

そろばんがディジタル形の計算機のめばえであるとするならば，その歴史は実に5000年も昔にさか上ぼる。その後，近世にいたり，機械式の計算機がはじめてパスカル(1)によって作られた。この計算機はさらにライプニッツ(2)によって乗除算も可能なものに改良された。この系列に連なる計算機は，その後，卓上計算機として発展し現在にいたっている。



第4.1図　ディジタル電子計算機発達の流れ

今日のディジタル形自動計算機の開祖は，イギリスのバベジ(3)といわれている。彼は1812年に階差エンジンを考案したが，これは基本的には加算機であり，多項式を計算するのに便利であった。さらに1830年代になりバベジは，当時すでに出現していたせん孔カードを利用した織機の原理を取り入れて，新しい機械，解析機関の作成に取り組んだ。しかし，当時の技術をもってしてはバベジの夢を実現することはできなかった。

バベジの考えた解析機関は，演算機能のほか記憶機能も持っており，せん孔カードから計算手順を読み取り，制御を行ない，その計算結果を自動的に印刷するとい

(1)　B. Pascal　　(2)　Leibnitz　　(3)　C. Babbage

う今日の電子計算機の持つ基本的機能をそなえていたのである。

このバベジの構想が初めて実現されたのは，アメリカのハーバート大学が 1944 年に完成した，電磁的な継電器を用いた Harvard Mark 1 においてであった。

真空管を用いた電子計算機の最初のものは 1946 年 ENIAC(1) である。その処理時間が加減算で 200μs であった。これは当時として驚くべき速度であったが，しかし，その記憶容量はわずか百数十けたにすぎず，プログラムもプログラムによる配線が行なわなければならなかった。

同じ年に，フォン・ノイマンは第2章で述べたプログラム記憶式計算機についての構想を明らかにした。そしてこれを具体化した EDVAC(2) が 1949 年に発表された。これは今日の形の電子計算機の元祖というべきものであった。

1951 年になると最初の商用電子計算機 UNIVAC 1 が現われた。この頃から1957，8年ごろまでの電子計算機は，演算装置や制御装置に真空管を用いていた。また，記憶装置にも水銀遅延線（電気信号を一度超音波振動に変えて水銀中を通し，その遅れ時間だけ記憶を行なうもの）などが用いられていた。この時代の電子計算機は，人手で行なうには困難で複雑な計算を行なうために使用され，したがって科学計算用が主であった。この時代の電子計算機を第一世代の電子計算機と呼んでいる。

1950年代の末ごろから，電子計算機は第二世代にはいったといわれている。論理回路には真空管に代わってトランジスタやダイオードなどの半導体素子が使用されるようになり，また，記憶装置にも磁心が使われるようになった。その結果，処理速度が上がり，記憶容量が増大するとともに，装置は小形になり，信頼度も著しく向上した。そして単に複雑な計算処理を行なうだけでなく，大量のデータを処理する事務用としての用途が開けたのである。

事務を機械化しようという動きはこれ以前にもなかったわけではない。たとえば各種のデータをカードにせん孔し，統計を作成する方法はすでに 1890 年にホレリス(3)により考案されていたが，その後せん孔機，分類機，会計機などの多くの事務

---

(1)　Electronic Numeric Integrator and Computer の略
(2)　Electronic Descrete Variable Automatic Computer の略
(3)　Herman Hollerith

用機械が作られ，せん孔されたカードをもとに，分類，照合を行なったり，計算を行なったり，あるいは統計資料や報告書を作成したりするようになった。これをPCS(1)と呼んでいる。

第2世代の計算機は前に述べたさまざまな特徴を有しているが，同時に入出力装置の速度も著しく高められ，大量のデータを入出力する事務計算に電子計算機を適用することを可能にした。そして PCS をより能率の高いかつ必要に応じて複雑な処理も行なうことのできる電子計算機システムへと変えていくことになった。

1966年ころになると，集積回路を利用した電子計算機が現われ，第三世代を迎えた。この世代の電子計算機では，すでに事務用とか科学用とかの区別は失なわれ，プログラムの組み方によっていずれにも使用しうる汎用形であった。第三世代の電子計算機を特徴づけるものに，ファミリー化あるいはシリーズ化と呼ばれるものがある。これは小規模な電子計算機で最初出発し，その後処理する量が多くなって，より大規模な電子計算機に変更する際に，同じファミリーまたはシリーズのものなら，すでにあるプログラムを全面的に変更する必要もなく，また前からあった入出力装置がそのまま使用できるというものである。

この世代の電子計算機のもう1つの特徴は，オンライン・リアルタイム・システムへの適用である。この世代の電子計算機には，第3章で述べた割り込み処理やダイナミック・リロケーションや記憶保護などの機能が設けられ，より複雑な処理を行なうことが可能となった。

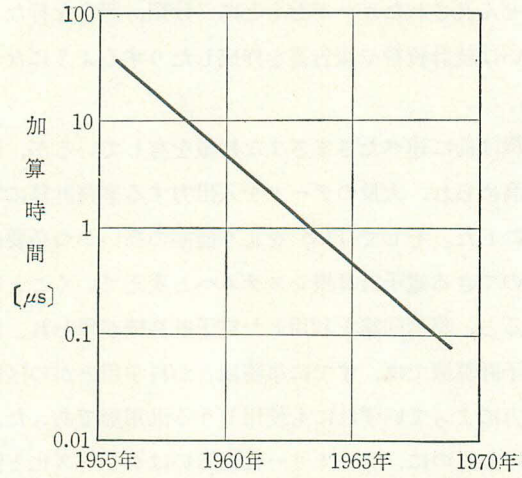
4.2 電子計算機の発達と今後の展望

4.2.1 処理速度と記憶容量

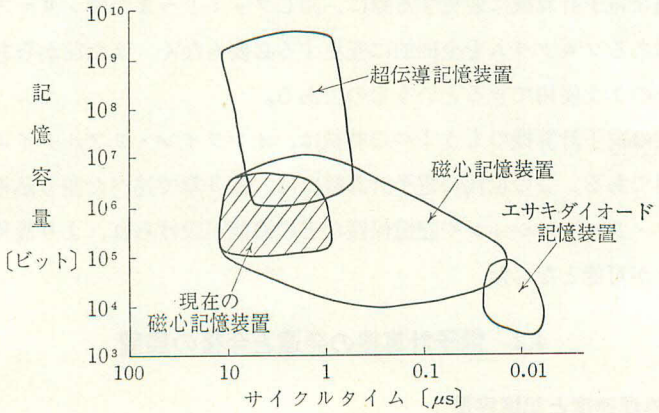
第4.2図は電子計算機の処理時間が短縮されてきた状況を示したものである。このような高速化を可能にした一つの原因は，論理回路に使用される部品が真空管からトランジスタやダイオード，さらに集積回路へと進歩してきたことである。

一方，電子計算機の処理速度を向上する上で，主記憶装置の高速化がかなり重要な要因となっている。すなわち，主記憶装置の速度は論理回路に比べればかなりお

(1) punched card system の略



第4.2図　電子計算機の処理速度の向上（加算時間）



第4.3図　記憶装置の容量とサイクルタイム

そく，処理速度は記憶装置の速さによって支配される場合が多い。

第4.3図に主記憶装置として今後使用されるであろうものを含めてその速度と記憶容量を示す。

主記憶装置の記憶容量の増大も電子計算機の処理能力の向上にとってきわめて重要である。高速で大容量の記憶装置の開発は今後も進められるであろうが，それと

は別に高速小容量から低速大容量まで記憶装置を何段階かに階層化し，それらをもっとも効率的に利用するくふうもなされている。

処理能力を上げるもう1つの方法として，電子計算機複合体を用いる方法があり，たとえば第3章で述べたマルチプロセッサ・システムのように，並列に処理を進めることによって実効的な処理速度の向上が図られている。こうした処理能力の向上は電子計算機の経済的な利用をもたらし，また，その新しい応用面を開くものとして，今後一層の努力が続けられていくであろう。

4.2.2 電子計算機と人間

電子計算機はぼう大なデータを正確に記憶する点において，また高速で計算を行なう点において人間とは比べものにならないほどの能力を持っている。しかしながら，創造力と判断力に関しては電子計算機はとうてい人間のまねをすることができない。それゆえ，電子計算機を利用する場合には人間と電子計算機が協調し合い，お互いの長所を生かしつつ両者が一体となって1つのシステムを作ることが望ましいのである。このようなシステムのことを**マンマシン・システム**(1)という。

マンマシン・システムの中では，人間と電子計算機の間の意志の疎通が十分行なわれていなければならない。このためには人間と電子計算機との間の通信，すなわち，マンマシン・コミュニケーションをいかに行なうかが問題である。これは，具体的にはデータの入出力をどうするかということになるであろう。

人間に理解しやすい図形や文字や音声を用いて出力するのは，機械にとってそれほどむずかしくはないが，図形，文字，音声などをデータとして入力するのは，きわめてむずかしい問題である。このような形のデータを機械が識別することを**パターン認識**(2)といい，3. 4 で述べたようにきわめて簡単な文字の識別など，すでに実用化されているものもあるが，さらに自由な形で描かれたパターンを識別しうるようになるのはまだ先のことであろう。

プログラム言語もマンマシン・コミュニケーションのための重要な問題の1つである。この言語は，人間にとってできるだけ使いやすい言語であることが望ましい。たとえば人間どうしが会話を行なうと同じように，電子計算機と人間が交互に

(1) man-machine system (2) pattern recognition

情報を交換しながら仕事を進めていくシステム（タイムシェアリング・システムではこのようなことはひんぱんに起きる）では，そのための簡潔で理解しやすい言語が必要であり，会話用言語として開発が進められている。

4.2.3　電子計算機の応用

初期の電子計算機は科学技術用が主体であったが，今日では，はるかに広い分野において使用されるようになってきており，単なる事務処理のような補助的作業を行なうにとどまらず，電子計算機が本来の業務の中に完全に密着している例も数多くみられる。そのもっともよい例は **3.1.6** で述べた自動制御用のシステムであろう。鉄鋼，石油，化学工場などにおいては，生産工程の自動制御用として電子計算機はきわめて重要な位置を占めており，その他，自動車工場における組み立て工程の制御，利水，給電の制御，あるいは交通制御などにも使用されている。また，予約システムなども電子計算機が本来の業務そのものを実行している例である。

企業にとって，長期計画を立て，市場調査を行ない，販売状況を知り，資金計画を立て，適切な購売を行ない，生産状況を管理し，製品の品質管理を行なうなどの業務を最適な方法で行なうことはきわめて重要であるが，そのためには電子計算機は欠くことのできない存在である。

銀行などにおいては，すでに為替（かわせ）の交換，預金など各種の業務にオンライン・システムが導入されているが，将来は商品を買うとその代金は銀行の電子計算機に記憶されている預金高から自動的に差引かれるという，いわゆるキャッシュレス時代の到来が予想されている。

電子計算機の教育面への利用も今後発展が期待される分野である。たとえば，多数の端末装置がオンラインで電子計算機と接続されており，各端末装置の前にすわった生徒が電子計算機と会話しながら，1人1人の能力に応じた教育を受けるようなことも可能となろう。このようなシステムを CAI[1] といっている。

このほか電子計算機は，病気の診断，新聞の印刷，人事管理などきわめて広い分野に応用されていくものと考えられるが，最後に特に重要な応用面として MIS と NIS についてふれておきたい。

（1）　computer aided instruction

MIS(1)(経営情報システム)とは，経営のすべての階層，たとえば社長や重役などのトップレベルから，実際に生産にたずさわっている部門にいたるまで，その業務を行なうのに必要な情報を必要な時期に提供する効率的な仕組みのことである。すなわち，このシステムでは，電子計算機は個々の業務だけを扱うのではなく，それらを総合し，情報を一元化している点に特徴がある。このようなシステムでは，企業全体を動かすものとして電子計算機の価値はきわめて重要なものとなろう。

このように，情報の総合化をさらに国家的なレベルにまで高めたものがNIS(2)といわれるものである。

このように，電子計算機はより広くまたより重要な使われ方をするようになってきており，今後ますますわれわれの社会に密着して，必要欠くべからざる存在となっていくことは間違いないものと思われる。

4.3 摘　　要

1. 今日の形の電子計算機は EDVAC を元祖として，その後，真空管を用いた第一世代の電子計算機，トランジスタを使用する第二世代の電子計算機，さらには IC を用い，汎用形でオンラインにも適した第三世代の電子計算機へと変わってきた。

2. 電子計算機は今後処理効率の一層の向上が予想されるほか，人間との通信もより円滑になり，多様な応用面が開かれ，よりわれわれの生活と密着したものになると考えられる。

4.4 練 習 問 題

1. 関連ある言葉を線で結べ。

(1) フロント・エンド	(a) マンマシン・コミュニケーション
(2) ダイナミック・リロケーション	(b) フェイルセーフ
(3) コスト・パフォーマンス	(c) 通信制御装置
(4) パターン認識	(d) ロールイン，ロールアウト
(5) デュアル・システム	(e) グロッシュの法則

2. オンライン・リアルタイム・システムの具体例を挙げよ。

---

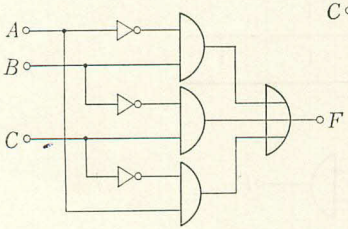
(1) management information system の略
(2) national information system の略

練　習　問　題　の　答

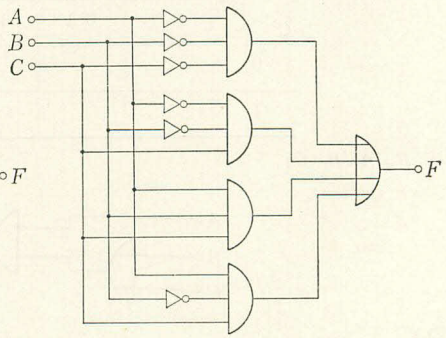
第1章

1. (1) 1110,　(2) 100000011,　(3) 101101010,
2. (1) 17,　(2) 57,　(3) 42
3. (1) 0001 0100,　(2) 0010 0101 1001,　(3) 0011 0110 0010,
4. (1) 1000,　(2) 10011,　(3) 1000000,
5. (1) 110,　(2) 10,
6.

(1)

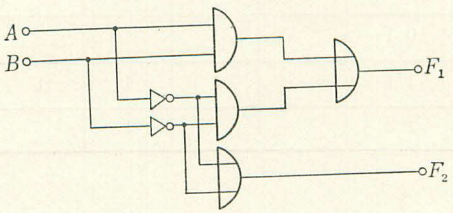


(2)



7. $F_1=A'\cdot B'\vee A\cdot B'$

$F_2=A'\vee B'$



8. (1) $F=A\cdot B\vee C$

真 理 値 表

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(2) $F=(A'\vee B)\cdot C'$

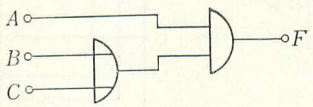
真 理 値 表

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

9. (1)

C \ AB	0 0	0 1	1 1	1 0
0			1	
1			1	1

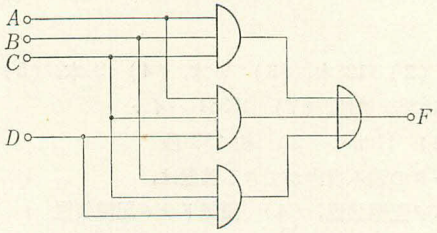
$F=A\cdot(B\vee C)$



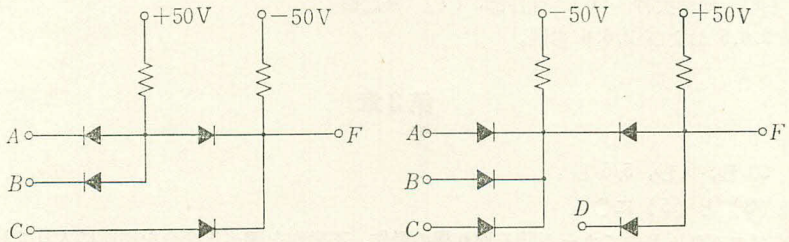
(2)

CD \ AB	0 0	0 1	1 1	1 0
0 0				
0 1				
1 1		1	1	1
1 0			1	

$F=A\cdot B\cdot C\vee A\cdot C\cdot D\vee B\cdot C\cdot D$



10. (2)



11. 負論理の nor 回路

入出力電圧

入力〔V〕			出力〔V〕
A	B	C	F
0	0	0	10
0	0	10	10
0	10	0	10
0	10	10	10
10	0	0	10
10	0	10	10
10	10	0	10
10	10	10	0

真理値表

入力			出力
A	B	C	F
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	1

12. $F=(A\vee B)\cdot C'$

第2章

1. **2.2.3** 参照。
2. (1) 128 本, (2) 128本, (3) 9 本, (4) 9 本, (5) 147456 個, (6) 0 番地～16383 番地, (7) 16384 バイト
3. (1) 6ms, (2) 110ms, (3) 約 233 秒
4. ランダムにデータの読み書きができる装置は, (2) 磁気ディスク記憶装置, (4) 磁気ドラム記憶装置
5. **2.6.2** 参照。
6. (1) 記憶装置, (2) 命令取り出し, (3) 制御装置, (4) 命令レジスタ, (5) 命令実行, (6) 操作部, (7) 解読器
7. **2.6.5** および **2.6.6** 参照。

第3章

1. 20 秒, 5 秒, 5/6 秒
2. (3) と (5) に○
3. プログラムB中にカード読み取り機の動作, 不動作を調べる命令を随所に入れておき, 動作中ならBの処理を続け, 不動作ならAの処理に移るようにしておけばよい。しかし, このような命令をBの中にひんぱんに書くことはわずらわしく, かつBの処理時間も長びく。

第4章

1. (1)-(c), (2)-(d), (3)-(e), (4)-(a), (5)-(b)

索　　引

ア

IBG …… 145
RTL回路……97
and ゲート……24
and 回路……24
and 条件……22
アクセス・タイム…… 124,151
アセンブリ語…… 181
アドレス…… 126
アドレス・レジスタ…… 131
合鍵方式…… 242
誤り制御…… 258

イ

EOT …… 145
EDVAC …… 284
EDPS …… 216
インデックス・マーク…… 158
インバータ……25
位相変調…… 250
位相変調方式…… 140
一括処理…… 217
印字部…… 262

エ

NIS …… 289
NRZ …… 136
NRZI 方式…… 138
npn 形トランジスタ……81
n 形半導体 ……79
エミッタ接地形……84
演算回路…… 3
演算装置…… 107,110,177

オ

or ゲート ……25
or 回路 ……25
or 条件 ……23
オーバーヘッド…… 226
オフライン…… 212
オープン・ループ…… 228
オペレーティング・システム…… 230,239
オンライン…… 212
オンライン・システム…… 216
オンライン・リアルタイム・システム… 216
オンライン・リアルタイム処理…… 160
応用プログラム…… 240
送りレジスタ…… 36,37
押しボタンダイヤル電話機…… 277

カ

カードせん孔機…… 165
カード読み取り機…… 163
カルナウ図……70
回路部品……78
書き込み…… 118
書き込みヘッド…… 136
書き込み電流…… 120
解読器……54
活性領域……83
管理プログラム…… 240
簡素化（カルナウ図の）……31

キ，ギ

キー・ボード…… 261

キー・ボード・プリンタ…… 261
奇偶検査…… 130
記憶回路…… 3
記憶装置…… 109
記憶保護…… 241
記憶容量…… 286
基数…… 5
機械語…… 120
禁止電流…… 120
偽……65
逆方向……79
行印字機…… 171

ク，グ

クロック・パルス…… 137
クローズドループ…… 228
グロッシュの法則…… 223

ケ，ゲ

けた上げ…… 5
けた上げ先取り方式…… 179
けた上げ記憶回路……35
けた送り……36
けた送り信号……37
けん盤テープせん孔機…… 171
けん盤部…… 261
結合の定理……68
減算回路……40

コ

コア・プレーツ…… 125
コード…… 3
コスト・パフォーマンス…… 223
コレクタ電流……82
コンパイラ…… 181
光学マーク読み取り装置…… 264
光学文字読み取り装置…… 267

サ

サイクリック・コード…… 258
サイクル・タイム…… 124
再書き込み…… 119
3 レベル RZ 方式…… 139
3 増しコード…… 9
三角波状セレクタ・バー…… 261

シ，ジ

CRT 表示装置…… 275
シーケンシャル・アクセス装置…… 141
シリンダ…… 157
シングル・ギャップ・ヘッド…… 143
しゃ断状態……93
しゃ断領域……83
支援プログラム…… 240
主記憶装置…… 107
主制御命令…… 241
周波数帯域…… 255
周波数変調…… 249
周辺装置…… 108
集積回路……95
出力回路…… 3
出力装置…… 107, 111, 275
出力変換回路…… 3
処理速度…… 285
消去ヘッド…… 136
真……65
真空コラム方式…… 147
真理値表……21
振幅変調…… 249
磁気コア…… 114
磁気テープ装置…… 140
磁気ディスク記憶装置…… 154

磁気ドラム…… 149
磁気ドラム記憶装置…… 148
磁気ヘッド…… 134
磁　心…… 114
磁心マトリクス…… 125
磁心記憶装置…… 114
実時間処理…… 215
10進法…… 4
除算回路……47
乗算回路……43
情　報…… 105
巡回制御…… 258
順方向……79

ス，ズ

スイッチング・トランジスタ……83
スタック…… 126
スループット…… 226
数字表示窓…… 1
図形表示装置…… 274

セ

セクタ・ディスク…… 158
セット……19
セット信号…… 19, 95
セット状態……94
正理論……87
制御回路……58
制御装置…… 107, 111, 180
制御卓…… 176
整流作用……79
絶対アドレス…… 237
全加算器……29
全二重通信…… 257
全二重通信方式…… 257

ソ，ゾ

ソフトウェア…… 106
ソフト・コピー…… 260
相対アドレス…… 237
増幅作用……83

タ

タイプ・ホイール…… 263
タイプ・シリンダ…… 262
タイミング制御…… 257
タイムシェアリング・システム…… 223
多重プログラミング…… 229
多重プロセッサ…… 246
多重プロセッサ・システム…… 246
多重プロセッシング…… 246
単向通信方式…… 256
端末装置…… 258
ダイオード……79
ダイナミック・リロケーション…… 238

チ

チャネル…… 202
遅延回路……38
逐次制御カウンタ…… 190
逐次制御方式…… 190
中央処理装置…… 108, 217
調歩式…… 257
直流電送…… 248
直列加算方式……33
直列転送法…… 252
直列伝送…… 252

ツ

通信制御…… 242
通信制御装置…… 217

通信速度…… 255
通信方式…… 255

テ，デ

テープ読み取り機…… 169
テープせん孔機…… 170
DTL 回路……97
ディスク・パック…… 154
データ…… 104
データ通信…… 217
データ収集システム…… 226
データ分配システム…… 227
ディレイドタイム…… 215
ディレイドタイム処理…… 215
デュアル・ギャップ・ヘッド…… 142
デュアル・システム…… 244
デュプレックス・システム…… 245
伝送制御…… 257
電子データ処理システム…… 216
電子計算機複合体…… 244
電子式卓上計算機…… 1
電流一致方式…… 121
トラック…… 142
特権命令…… 241
ド・モルガンの定理……69
同期式の計算機……39
動的再配置…… 238

ナ

nand回路（トランジスタの）……91
内部コード……3,54
内部抵抗……83

ニ

2 out of S コード…… 9
2 ギャップ・ヘッド…… 142
2 レベル RZ 方式…… 138
2 に対する補数……41
2 進 1 けた……28
2 進化10進法…… 9
2 進法…… 5
2 線式…… 256
入出力装置…… 108
入力回路…… 3
入力装置…… 107,108,276
入力変換回路…… 3

ノ

nor 回路（トランジスタの）……89
not 回路……25
能力の比…… 223

ハ，バ，パ

Harvard Mark I…… 284
ハードウェア…… 106
ハード・コピー…… 260
ハンドリングタイム…… 230
8 進法…… 5
破壊読み出し…… 118
半加算器……29
半二重通信方式…… 257
バイト…… 126
バベジ…… 283
バックエンド…… 217
バックグラウンド処理…… 218
バッチ処理…… 217
バンド…… 151
パターン認識…… 287
パスカル…… 283
パリティ・チェック…… 130

ヒ，ビ，ピ

ヒステリシス特性曲線…… 115
否定の定理……68
否定回路……25
否定回路（トランジスタの）……88
非同期式……39
非破壊読み出し…… 136
ビット直並列形式…… 151
ビット直列形式…… 151
ビット並列形式…… 151
p形半導体……79
pnp形トランジスタ……81
PCS…… 285

フ，ブ，プ

ファミリー化…… 285
フェイル・セーフ…… 231
フェイル・ソフト…… 231
フォン・ノイマン…… 284
フリップフロップ回路……19
フロントエンド…… 217
符号器……54
負理論……87
浮動ヘッド…… 153
復　調…… 248
ブール代数……31
ブロック…… 145
部分積……16
分配の定理……68
プリンタ…… 262
プログラミング…… 104
プログラム…… 104
プログラム記憶式計算機…… 104
プログラム言語…… 181

ヘ，ベ

並列転送法…… 252
並列加算方式……33
並列伝送…… 252
返送照合方式…… 258
変換の定理……68
変　調…… 248
変調速度…… 254
変調方式…… 249
変復調装置…… 249
ベースバンド伝送…… 248
ベース・レジスタ…… 237
ベース電流……82

ホ，ボ，ポ

ホレリス…… 284
ホレリス・コード…… 163
補助記憶装置…… 107, 133
補数器……41
補　正……12
方形歯状セレクタ・バー方式…… 261
飽和回路……93
飽和状態……93
飽和領域……83
BOT …… 145
ボ　ー…… 252
ポーリング・セレクション…… 243

マ

マーク・シート…… 264
マスタ・ファイル…… 221
マン・マシン・システム…… 287

メ，モ

メッセージ…… 219

メモリ・インタリーブ…… 230
メモリ・レジスタ…… 131
命　題……65
命令レジスタ……58
文字表示装置…… 271

ユ，ヨ

UNIVAC 1 …… 284
呼び出し応答方式…… 243
呼び出し時間…… 151, 124
読み出し…… 116
読み出し電流…… 118
4線式…… 256

ラ

ライトペン…… 276
ライプニッツ…… 283
ランダム・アクセス記憶装置…… 217
ランダム・アクセス装置…… 149

リ

リアルタイム…… 215
リアルタイム処理…… 215
リセット……19
リセット状態……94
リセット信号…… 20, 95
リモートバッチ処理…… 218

ル，レ

累算器……36
レコード…… 145
レジスタ……18
連送方式…… 258

ロ

ロールアウト…… 225
ロールイン…… 225
ロード…… 189
論　理……60
論理回路……60
論理回路（トランジスタの）……88
論理学……65
論理式……60
論理積……23
論理積回路…… 24, 60
論理代数……31
論理和……24
論理和回路…… 25, 62

ワ

割り当て（主記憶装置の）…… 235
割り込み…… 229, 231
割り込み信号…… 232
割り出し…… 229

昭和45年 9 月25日　初版発行　定価 1,000円
昭和46年 6 月21日　再版発行

電子計算機の原理と構造

（ハードウェアの知識）

著作者　電気学会通信教育会
発行者　巽　良知
印刷所　大日本法令印刷株式会社
製本所　染野製本所

発行所　社団法人 電気学会
郵便番号 100　東京都千代田区有楽町1の3
電話東京 (213) 4891～3
振替口座東京720（電気学会大学講座）

取次販売所　株式会社 学献社
東京都文京区小石川3丁目1番3号（伝通院ビル内）
郵便番号112　電話 (813) 7362～3　振替東京13152

乱丁落丁の際はお取替えします

電気学会の情報工学関係図書

事務系・技術系を問わず，系統的知識を得るための入門編。
内容は，一部高校・企業などでわかりやすさを実験済み。
第2種情報処理技術者認定試験科目をカバー。

電子計算機のプログラミング

——ソフトウェアの知識——

A5判　320頁
定　価　1000円

執筆委員　佐　藤　達　男（東海大学教授）

FORTRAN, ALGOL, COBOL, PL/I 等を身近な例題によって具体的に理解させる。コーディングシートの書き方をはじめ，プログラムは，実際に電子計算機を通したものを逐一掲げ，流れ図，結果を添えてある。また，ファイル，チェックについても解説。

情報処理のための数学

A5判　約300頁
予　価　1000円

執筆委員　三重野　博　司（東京理科大学教授）

全体を2編に分け，第1編は，電子計算機を理解あるいは利用するための数学的な考え方が中心で，数式は余り使っていない。記号と集合，記号と記号の関連づけ，人間の認知しうる空間等を扱っている。数学的基礎知識を忘れた人は第2編で復習すればよい。

電子計算機一般

A5判　183頁
定　価　290円

執筆委員　篠原　靖市（都立商業教育実習所）他

初心者に，電子計算機のハードウェアからソフトウェアまでの全システムを教える入門編

電子計算機

A5判　302頁
定　価　900円

執筆委員　工博　駒　宮　安　男（電子技術総合研究所）

電子演算工学概論

A5判　442頁
定　価　1800円

執筆委員　工博　木　村　久　男（成蹊大学）他

測定値の統計的処理

A5判　222頁
定　価　660円

執筆委員　工博　石　井　　泰（東大宇研）他

大学院講座・拡張アルゴル

A5判　214頁
定　価　850円

執筆委員　奥　村　利　夫（高千穂交易）

東京工大　山口晶一郎　執筆
基礎電磁気学
266頁　700円

電荷と電界，電位，種々な帯電体による電界，静電容量，誘電体，電流と抵抗，磁界，電磁誘導，インダクタンス，変動電流回路，磁性体，電磁波

千葉大学　小郷　寛　執筆
交流理論
332頁　820円

正弦波電圧，正弦波電流，インピーダンス，記号法，交流回路，交流電力，多相交流，ひずみ波交流，過渡現象

千葉大学　小郷　寛　執筆
回路網理論
188頁　460円

直流電圧電流，交番電圧電流，ベクトル記号法，交流回路，一般線形回路網，2端子網，4端子網，多相交流，ひずみ波交流，分布定数回路

東京大学　西野　治　執筆
電磁気計測
280頁　700円

計測一般，電気計器，電磁気測定，電気応用計測，電子計測

埼玉大学　天野寛徳　執筆
電気機械工学
434頁　990円

直流発電機，直流電動機，変圧器，誘導機，同期機，交流整流子機，交直流変換装置

横浜国立大学　弘山尚直　執筆
発変電工学
372頁　820円

総論，水力発電，汽力発電，汽力発電以外の火力発電，電気設備，計画設計と運転保守

東京大学　中路幸謙　他執筆
電気応用
289頁　870円

電燈照明，電熱，電気鉄道，応用電気化学，各種の電気応用

早稲田大学　加藤一郎　執筆
制御工学
298頁　750円

自動制御の特質，伝達関数とブロック線図，過渡応答，周波数応答，安定判別，性御性能，むだ時間をもつ系，サーボ機構，プロセス制御

東海大学　西川甚太　執筆
工業計測
267頁　980円

変位・ひずみ・回転速度・厚さの測定，液位・圧力・差圧・粘度の測定，流量測定，成分分析，温度の測定，指示計，記録計，調節計，計装等

電気通信大学　関　英男　執筆
電子工学概論
266頁　650円

電子とその作用，電子管，半導体素子，電子回路，電気通信，電子応用

上智大学　吉田裕一　執筆
電子回路
320頁　800円

電子装置，増幅回路，整流回路，発振回路，波形処理・発生回路，変調・復調回路，電界効果トランジスタとその回路

通商産業省　冨田弘平　他執筆
電気施設管理と法規解説
（再改訂）346頁　1500円

電力需給，電力施設の建設計画，電力系統の運用，電気料金と電気事業会計，電気法規一般，技術基準，規格

電気学会の既刊書

「原子力からエレクトロニクス」まで，定評ある電気学会の工学書を，あなたの能力開発に！

電気磁気学（改訂）	980円	送電工学（改訂）	1100円
電気回路論（改訂）	1300円	配電・屋内配線（再改訂）	580円
過渡現象論	800円	電気材料（改訂）	800円
放電現象	550円	電気材料便覧	1300円
原子物理学	500円	高電圧工学（改訂）	980円
基礎電子工学	650円	電気鉄道	800円
電離気体論	1100円	電気鉄道ハンドブック	1800円
物性論	550円	電熱工学	600円
電気応用計測	760円	照明工学	800円
電気計測器	1200円	電気化学	950円
電気測定法	950円	産業と電気	900円
測定値の統計的処理	660円	電動力応用工学	1400円
水銀整流器	660円	電子応用	950円
特殊機器	700円	家庭電機工学概論	1600円
遮断器・開閉器	900円	電気施設管理	700円
電機設計概論（改訂）	550円	電気法規技術基準解説	1000円
電気機器工学 I	900円	有線通信	960円
電気機器工学 II	550円	無線通信	860円
直流機	550円	トランジスタ	700円
変圧器・電導機 交流整流子機	1000円	自動制御	750円
同期機	1300円	電子管	780円
静止電力変換装置	700円	電子計算機	900円
水力発電（改訂）	1150円	電子演算工学概論	1800円
火力発電	1000円	半導体電子回路	900円
原子力発電（改訂）	1500円	大学電気数学	750円

電気学会のやさしい数学入門編

全問題について解法を示した学習の手引を
各巻末に収録。

執筆委員　管　原　正　巳（科学技術庁）

電　気　数　学　Ⅰ

A 5判・本文 278頁
学習の手引　67頁

定 価 1,100 円

代数入門，幾何入門，数式の変形，二次方程式，方程式・不等式の解法，複素数，指数関数と対数関数，計算図表，空間図形，微小量の取扱

電　気　数　学　Ⅱ

A 5判・本文 224頁
学習の手引　89頁

定 価 950 円

三角関数，平面座標幾何，立体座標幾何，関数の微分変化，微分法，極大極小，指数関数と対数関数の微分，近似式と展開式，微分の逆演算

電　気　数　学　Ⅲ

A 5判・本文 336頁
学習の手引　101頁

定 価 1,350 円

微分法と積分法，積分法，積分の応用，数値積分法，展開式，微分方程式，偏微分，重積分，確率，統計

中学程度の数学知識を整理しながら，知らず知らずに
技術数学が身につく，ソフトムードの数学書。

電気学会工高講座

電気工学・電子工学の技術者になるための入門編。

A５判　箱　入

電気理論Ⅰ（付学習指導書）	定価 800円	電気材料（付学習指導書）	定価 550円
電気理論Ⅱ（付学習指導書）	定価 500円	自動制御（付学習指導書）	定価 700円
電気計測（付学習指導書）	定価 750円	電子工学（付学習指導書）	定価 800円
電気機器Ⅰ（付学習指導書）	定価 800円	電子現象（上製，カバー付）	定価 200円
電気機器Ⅱ（付学習指導書）	定価 550円	電子回路工学（上製，カバー付）	定価 980円
発送配電Ⅰ（付学習指導書）	定価 750円	電子機器（上製，カバー付）	定価 880円
発送配電Ⅱ（付学習指導書）	定価 800円	電気法規概説（上製，カバー付）	定価 250円
電気応用（付学習指導書）	定価 800円	電気工学概説（上製，カバー付）	定価 900円

電気学会の電気実験集

実験項目・目的，原理，実験順序，実験結果等がわかりやすく解説されています。

B５判　箱入ビニールカバー付

電気実験（電子編）	123頁	定価 400円
電気実験（基礎・計測編）	74頁	定価 360円
電気実験（電気機器・電力編）	113頁	定価 400円

電気学会の教育用スライド

完成された外観のみならず，その製造過程，工事過程，試験など，得難い写真を集め，随所に線図を加えて説明に使用したスライドシリーズ（35ミリ判）で，各専門のメーカの協力を得て製作。

（詳細な説明用の**スライドガイド**付）

電気機器Ⅰ **変　圧　器**	【概要】 変圧器の外観，鉄心，巻線，ブッシング，冷却方式，タップ切換，油劣化防止方式，保護装置，絶縁，輸送，試験その他，構造，各種の変圧器　—日立協力— 61コマ（カラー6コマ）定価 3,000円
電気機器Ⅱ **誘　導　機**	【概要】 三相誘導電動機の構造，誘導電動機用制御機器，単相誘導電動機，誘導電動機の応用製品，誘導電圧調整器　—東芝協力— 67コマ（カラー10コマ）定価 3,500円
電気機器Ⅲ **同　期　機**	【概要】 同期機の外観，水車発電機の構造，タービン発電機の構造，その他の発電機（高周波発電機，発電電動機，短絡発電機）—三菱協力— 67コマ（カラー7コマ）定価 3,500円
電気機器Ⅳ **直　流　機**	【概要】 直流機の外観，直流機の構造，界磁，整流子，ブラシおよびブラシ保持器，直流電動機の始動器，特殊直流機，直流機の応用　—富士協力— 57コマ（カラー3コマ）定価 2,500円
発変電Ⅰ **水　力　編**	【概要】 発電方式，発電所の形式，ダム，取水口，導水路および水そう，水圧管路，水車，発電機，変圧器，監視制御装置，揚水式発電所　—電発協力— 79コマ（カラー9コマ）定価 5,000円
発変電Ⅱ **火　力　編**	【概要】 火力発電の概要，燃料の輸送と貯蔵，ボイラ，燃焼装置，通風装置，集じん・灰処理装置，タービン，復水器，給復水系統設備，水処理，タービン発電機 —東電協力—89コマ（カラー74コマ）定価 6,000円
発変電Ⅲ **変　電　編**	【概要】 変電所の概要，変電所の種類，変圧器，電圧調整器，開閉設備，保護装置，変電所の災害対策・工事・保守など　—東電協力— 64コマ（カラー1コマ）定価 3,500円
送配電Ⅰ **送　電　編**	【概要】 支持物，がいし，がいし装置，電線，架空送電線路の建設，保守作業，電力ケーブルの種類，ケーブルの布設方式，マンホール，管路，その他　—東電・関電工協力—　104コマ（カラー52コマ）定価 7,500円
送配電Ⅱ **配　電　編**	【概要】 配電線路（架空配電線路の構成，架空電線路の建設，地中電線路，他），屋内配線（絶縁電線の接続，接地工事，がいし引き工事，金属管工事，その他）—東電関電工協力—　109コマ（カラー21コマ）定価 8,000円

学　修　票

金沢大学工学部

昭和44年度入学	昭和45年度	前期 / (後期)	
工学部	精密工学	学科	
氏名	亀田直身 (亀田)	番号	10
科目名	精密加工学 II	必・選別	必修 / (選択)
担当教官名	本多 教官	単位	2

評価	優　良　可　不可	教官印	本多

(注)1. 太線以外の各欄は学生がペン書で記入する。
2. 学生は毎学期講義開始3週間以内に夫々担当教官に提出すること。
3. この学修票を提出しないと単位の認定を受けることができない。
4. 1科目2教官以上が担当している場合には教官名欄に教官名を列記して何れかの教官に提出する。

¥ 1000.